Employee Fraud Detection under Real World Conditions

DOCTORAL THESIS

FOR THE DEGREE OF A
DOCTOR OF INFORMATICS

AT THE FACULTY OF ECONOMICS,
BUSINESS ADMINISTRATION AND
INFORMATION TECHNOLOGY
OF THE
UNIVERSITY OF ZURICH

by
JONAS LUELL
from
SWITZERLAND

Accepted on the recommendation of

Prof. Dr. A. Bernstein
Prof. Dr. H. Geiger

2010

The Faculty of Economics, Business Administration and Information Technology of the Univeristy of Zurich herewith permits the publication of the aforementioned dissertation without expressing any opinion on the views contained therein.

Zurich, April 14. 2010

The Vice Dean of the Academic Program in Informatics: Prof. Dr. H. C. Gall

Acknowledgements

As no one reads the acknowledgements unless they are very short: Thanks to everyone who supported me, at the university, at the collaborating financial institute, and last but not least at home. Avi, Jiwen (and everyone else in the research group), Eva, Ueli, Susi, Claudia.... : Thank you for giving me this opportunity!

I dedicate this thesis to my dad.

Abstract

Employee fraud in financial institutions is a considerable monetary and reputational risk. Studies state that this type of fraud is typically detected by a tip, in the worst case from affected customers, which is fatal in terms of reputation. Consequently, there is a high motivation to improve analytic detection. We analyze the problem of client advisor fraud in a major financial institution and find that it differs substantially from other types of fraud. However, internal fraud at the employee level receives little attention in research. In this thesis, we provide an overview of fraud detection research with the focus on implicit assumptions and applicability. We propose a decision framework to find adequate fraud detection approaches for real world problems based on a number of defined characteristics. By applying the decision framework to the problem setting we met at Alphafin the chosen approach is motivated. The proposed system consists of a detection component and a visualization component. A number of implementations for the detection component with a focus on tempo-relational pattern matching is discussed. The visualization component, which was converted to productive software at Alphafin in the course of the collaboration, is introduced. On the basis of three case studies we demonstrate the potential of the proposed system and discuss findings and possible extensions for further refinements.

Contents

1

Introduction

This thesis studies the application of data mining and graph pattern matching techniques in compliance and risk related topics within a financial institution. The main focus lies on internal or employee fraud. Based on the findings, solutions to practical challenges in this problem field are proposed. A detection system, which has shown its potential under real world conditions, is introduced.

The research background originates from a collaboration with a major financial institution. To simplify matters, the collaborating company is called *Alphafin* within this thesis. Alphafin's fraud specialists, based on their expertise and prior tests, doubted common analytic fraud detection methods (like profiling) to be effective for the special case of internal fraud. A research project was started to further investigate the subject and to propose feasible solutions.
After concentrating on internal fraud, the focus was shifted to anti-money laundering related topics in a second stage, where the developed solutions were adapted and extended for the new application area. The findings of both stages are presented in this work.
Considering the common dichotomy of *IT* and *Business* in the financial sector, this thesis resides on the interface between the two positions. The design process is heavily influenced by both worlds on all abstraction levels. In particular, the developed algorithms are specialized to meet the practical requirements in a straightforward way.

In the course of this thesis, we tried to combine the perspectives of academia and industry on the one hand, and of informatics and business on the other hand. While being a computer science thesis, this work also intends to address the technically interested business expert. Although a very short introduction is given, basic knowledge of data mining and graph theory is assumed.

Relevant business information is given to the possible extent in the presence of non-disclosure agreements with Alphafin.

1.1 Motivation

The main background of this thesis is the mitigation of financial and reputational risk. Direct financial risk arises from losses caused by internal fraud. If a customer notices unjustified irregularities in his/her assets or if a bank is associated with internal fraud cases in the media, the effect can be a substantial loss in reputation. Losing reputation is losing trust, which is the foundation of banking business per se [Geiger, 2008]. Customers losing trust in their bank will withdraw their assets and terminate the business relation. It is therefore crucial for a financial institute to get employee fraud under control and, at best, give the impression that there is no such thing as internal fraud.

Improving a fraud detection system is typically about reducing a very high false positive rate[1]. False positives, which have to be investigated before being identified as what they are cause costs without direct benefit. Additionally, false alerts can upset the persons concerned. Nobody likes to be suspected. In the special case of internal fraud, these people are employees. Violating the mutual trust is dangerous for a company, as it can lead to a reduction in employee morale and loyalty. Minimizing this risk can be accomplished by identifying false positives before the involved person has to be informed or interrogated, e.g., by using a system which allows a holistic view on suspicious transactions. On the other hand, fraud detection systems may help to increase the expected risk for committing fraud. In "Crime and Punishment: An Economic Approach", the author suggests that the decision to commit a crime is based on a conventional cost-benefit analysis [Becker, 1968]. Knowledge or assumptions about sophisticated fraud detection systems and/or uncovered cases may therefore raise the inhibition threshold. However, at least at Alphafin, an analysis of known fraud cases questions the assumption of a rational cost-benefit decision as illegal behavior often seems to emerge from frivolity — and subsequent desperation (see section 2.1.2). Disclosure or concealment of the existence of fraud detection systems and uncovered cases is a classical tradeoff between deterrence and divulgement of informational advantage (see Figure 1.1).

The setup and configuration of detection systems in a company like Alphafin is a non-trivial

[1] = a very high number of "false alerts",i.e. instances which are reported by the system as suspicious but turn out to be unobjectionable in evaluation.

undertaking with high complexity. For example, new work flows have to be built and integrated, accounting for the highly confidential nature of the data. Adjusting and tuning system parameters requires comprehensive knowledge about the business and data. A common way to keep complexity and costs as low as possible is the limitation to — from an academic point of view — basic and well-known detection approaches, as e.g. plain threshold monitoring. This motivates the search for more expressive solutions which still meet the demands in terms of straightforwardness and feasibility.

For a profit-oriented company, evaluation of numerous novel approaches is typically out of scope. On the other hand, specialized commercial vendors may have the resources to do experimentation, but normally do not have access to the real data of their customers, which complicates the process. Academic research is a possible way to fill the gap and propose expressive but feasible approaches.

Figure 1.1: The employee information tradeoff

1.2 Problem Definition

Fraud detection suggests itself to be defined as an information-retrieval (*IR*) or *precision / recall* problem[2]. This requires the definition of the class of *relevant items*. One obvious possibility is to define relevant items as "fraud". At Alphafin, there is a number of reasons why this is problematic:

- Relevant activities may not be limited to explicit fraud, and a clear boundary between fraud and non-fraud may be hard to define. A certain behavior may be, for example, legal, but not compliant to company policies or, in an even weaker form, noneconomic and therefore undesirable. What is of interest for investigators may be fuzzy and may change over time or depend on variables which cannot be observed.

- Only a subset of the information which is necessary to clearly identify fraud is available in machine-readable format. This may be due to missing access authorization or to incompatible media formats. Investigators may proceed outside of the system as soon as a suspicion substantiates, for example in the form of interrogations and the review of free-text documents.

- As mentioned above, the human factor is crucial in the fraud finding process. The definition of "fraud" as relevant items mingles user experience with system accuracy.

An alternative is therefore the definition of a *relevant item* as "data of sufficient interest for the user" (*hot spots*) . Although this definition is less formal and quantification is difficult, it is more convenient for the actual problem. However, identification of precision and recall remains a challenge:

- The true recall is, by definition, unknown. To calculate recall, all fraud (or "interesting") data has to be known without exception, which is only possible if the problem we try to solve is already sorted out.

- Precision may be assessable, but its optimal value is not necessarily "one" as in typical *IR*-Systems. Users may want to see the "border" between interesting and uninteresting behavior. Even more, this border may become only clear in the process of analyzing the retrieved data. The system may be used not only for *hot spot* detection, but also (and at the

[2]For more information on information retrieval, see e.g. [Singhal, 2001]

same time) for learning about previously unknown structures in the data. Precision may not be an adequate quality measure in this case.

We therefore redefine the problem at hand more generally as *computer-assisted hot spot identification for domain experts*. The quality measure is the work efficiency and effectiveness when using the system based on expert assessment.

1.3 Hypothesis

Our main hypothesis is the following: “It is possible to combine and extend available data mining or pattern matching techniques to build a novel, feasible and valuable internal fraud detection system complementing existing solutions”. In the process of this thesis, design decisions led to a number of sub-hypotheses which will be mentioned in the appropriate passages.

1.4 Approach Overview

Our approach consists of two main components: a detection component and a result evaluation component.

The *detection component* provides the functionality to detect and retrieve hot spots in the data. This component is not a black box — the user is able to configure and adapt the detection algorithms. Different specialized implementations for this component are proposed in this thesis. The implementations vary in complexity and strategy for handling huge amounts of data, representing different points on the classical memory/runtime tradeoff .

The relational and temporal nature of the data and the properties of suspicious transaction patterns implicates that detected hot spots are extremely hard to investigate in the usual tabular data view. Therefore, an *evaluation component*, which relies heavily on visualization, was developed. It is tailored to the needs of internal fraud investigators, accounting for the special characteristics of internal fraud, which is described in closer detail in the following chapter. The insights from evaluation can be used to reconfigure the detection component, allowing a refining and learning process.

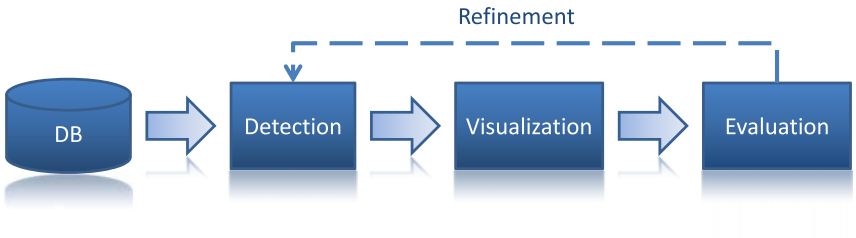


Figure 1.2: A very high level system overview

The evaluation component can be used without the detection component. In this case, hot spots are not identified automatically, but user-driven, for instance based on customer complaints or other hints. This mode is called *ad-hoc* search. The first stage of the collaboration with Alphafin focused exclusively on the design and development of the visualization component as it was required to get access to the data.

1.5 Contributions

This thesis is application oriented. Therefore the main contributions result from finding the best way to solve a real-world problem. We provide an answer to the following question: "How can existing work be adapted and applied to deliver a detection approach for the special case of employee fraud under real world conditions?"
Related contributions are:

- We analyze the potential of data mining and pattern matching techniques for internal fraud detection within a financial institution.
- We elaborate on the assumptions on which existing data mining solutions build and propose a framework to support the choice of a fraud detection approach in various settings.
- We discuss crucial issues of introducing fraud detection systems in financial institutes.
- We provide a fraud detection approach which stands the test of applicability under constraining conditions.
- We propose a number of potential paths building on this work and justify why preliminary steps were necessary.

1.6 Organization

This thesis is organized as follows.

First, we introduce the problem field and the attendant circumstances. We analyze the characteristics of internal fraud and related topics. The real world constraints, which played a major role in this work, are described. We then refer to related work with a focus on data mining and pattern matching approaches and discuss their applicability.

Second, we present our solution approach. We describe the system design and introduce different implementations for the detection component with their advantages and disadvantages.

Third, we discuss application aspects. The findings when evaluating on real data are given. For the sake of completeness, evaluation on synthetic data is delivered.

To conclude, we discuss the limitations of our approaches. Part of those limitations is supposed to be conquerable with future work. We introduce ideas for future research which in our opinion possess potential but were out of scope for this thesis.

2

Fundamentals, Premises and Related Work

2.1 Fundamentals in Risk and Compliance

Needless to state: fraud is a huge and very diverse area, which can be looked at from numerous possible views (e.g. from the social, legal, economic, or analytical perspective). We will keep the general information on fraud very short and focus on aspects which are of particular interest for this thesis. Consider the following possible definitions for the general term of fraud found in literature:

- "Intentional deception resulting in injury to another, as when a person makes false statements, conceals or omits material facts."[Fitch, 2006]
- "The abuse of a profit organizations system without that abuse leading necessarily to legal consequences"[Phua et al., 2005]
- "Criminal deception; the use of false representations to gain an unjust advantage"[Bolton and Hand, 2002]
- "In law, the deliberate misrepresentation of fact for the purpose of depriving someone of a valuable possession or legal right"[Encyclopedia Britannica, 2006]
- "A false representation of a matter of fact whether by words or by conduct, by false or misleading allegations, or by concealment of what should have been disclosed that deceives

and is intended to deceive another so that the individual will act upon it to her or his legal injury."[Jeffery Lehman, 2004]

The concepts of *deception* and *misrepresentation* at the damage of another individual are common terms for defining fraud — which is not surprising. As mentioned above, the term fraud denotes a wide and heterogeneous scope of activities. Prominent examples of fraud areas are

- Insurance fraud (e.g. fraudulent car or health insurance claims)
- Telecommunications fraud (phone cloning, subscription fraud)
- Investment fraud (pyramid schemes, insider trading)
- Employee fraud (falsification of balance sheets, embezzlement)
- Credit Card fraud (stolen or cloned credit cards)
- Retail fraud (forgeries, fake sales)
- Advance fee fraud (Nigerian money offer, lottery scam)
- Computer and internet fraud (Phishing, Spoofing,...)

As for example in computer and internet fraud, a clear differentiation between fraud areas and fraud instruments is not always straightforward. Instead of undertaking the definition of an exhaustive taxonomy of this broad field, we narrow our focus to known fraud schemes in the field of banking.

- Cheque fraud

  Cheques can be stolen, altered to an illegitimate payment recipient and higher transaction amount (adding a few digits) and/or provided with a forged signature or even be completely forged. The area of cheque fraud alone is a complex field, where, to our knowledge, detection is typically a highly manual process. Suspicious properties of hand- or machine-written cheques are recognized by trained human experts.

- Trading fraud

  An employee may trade sizeable assets on behalf of a customer or the bank without customer order. If invested money is lost, trading can become even more intense and aggres-

sive in the hope to cover the loss. This behavior led to some of the largest bank frauds ever detected [Clark and Jolly, 2008]

- Loan fraud

  Fraudulent loan applications as a form of external fraud can contain false information to hide financial problems. Also, employees may knowingly approve loans to accomplices who declare bankruptcy or vanish after receiving the money.

- Forged documents

  Forged documents (not only cheques and credit slips) can be used to trigger or cover illicit transactions.

- Bill discounting fraud

  This type of external fraud is on hand when a customer builds up confidence with a bank. Accomplices will readily and repeatedly pay bills issued by the customer (and raised by the bank). After successfully simulating reliable behavior, the customer requests that the bank settles its balance with the company before billing the customer. As soon as the outstanding balance between the bank and the company is large enough, the customer disappears with the money and his/her accomplices.

- Payment card fraud

  Payment cards can be stolen, duplicated or skimmed by various means. Obtaining the required information can for example be accomplished by manipulating ATMs.

- Identity theft

  This type of fraud works by obtaining information about an individual and using this information to apply for identity cards, accounts and credit in that persons name. Various ways of obtaining the required information are possible (e.g. indiscreet insiders or phishing).

- Trick fraud

  This summarizes fraud which is based on tricking legitimate account owners into paying money to the fraudster. Variants are simulating a “prime bank” with promising conditions, impersonating officials, forged emails and other pishing attacks.

- Computer fraud

  Technical attacks, in particular aimed at the identification and authentication mechanisms of e-banking solutions, is a type of relatively new bank fraud.

- Money laundering

  can also be seen as a special kind of bank fraud which aims at hiding the true (illicit) origin of funds.

Categories (Level 2)	Activities Examples (Level 3)
Unauthorized activity	Transactions not reported (intentional)
	Transaction type unauthorized (with monetary loss)
	Mismarking of position (intentional)
Theft and fraud	Fraud/ credit fraud/ worthless deposits
	Theft/extortion/embezzlement/robbery
	Misappropriation of assets
	Forgery
	Check kiting
	Smuggling
	Account take-over / impersonation etc
	Tax non-compliance/ evasion (willful)
	Bribes/ kickbacks
	Insider trading (not on firms account)

Table 2.1: *Basel II* categories and activities for event type category *Internal Fraud*

2.1.1 Internal Fraud

In the *Basel II* accord[1] [Basel Commitee on Banking Supervision, 2006], loss events from operational risk are broken down into seven general categories, of which one is *Internal Fraud*. The committee defines it as

"Loss due to acts of a type intended to defraud, misappropriate property or circumvent regulations, the law or company policy, excluding diversity / discrimination events, which involves at least one internal party "

The categories and activities examples for the event type category *Internal Fraud* defined in *Basel II* are listed in table 2.1.

Another definition is given by the *Association of Certified Fraud Examiners* (*ACFE*):

"The use of ones occupation for personal enrichment through the deliberate misuse or misapplication of the employing organizations resource or assets."

Extensive analysis in the field of occupational fraud from a business-oriented view has been done and published in particular by *ACFE* , whose annually *Reports to the Nation on Occupational Fraud and Abuse* are freely available [Association of Certified Fraud Examiners, 2008]. The authors propose an extensive occupational fraud and abuse classification system (see Figure 2.1). Another

[1]The Basel II accord forms recommendations on banking laws and regulations issued by the Basel Committee on Banking Supervision.

valuable information source is the series of papers in internal fraud management by the *Santa Fe Group*[2]. In our thesis, we will therefore just point out the most important findings in a nutshell before focusing on the characteristics of the internal fraud cases we analyzed at Alphafin.

[2] http://santa-fe-group.com/whitepapers

The *2008 Report to the Nation [Association of Certified Fraud Examiners, 2008]* based on data compiled from 959 cases of occupational fraud between January 2006 and February 2008 found that

- Participants of the study estimated that U.S. organizations lose 7% of their annual revenues to fraud, translating to approximately 994 billion dollars in fraud losses.
- The median loss in the analyzed cased is 175'000 dollars and one quarter of the cases involved losses of at least one million dollars.
- The typical fraud case lasted two years from starting the fraudulent behavior until its detection.
- The most common fraud schemes were corruption (27%) and fraudulent billing schemes (24%), while financial statement fraud was the most costly category with a median loss of two million dollars.
- Despite increased focus on anti-fraud controls in the wake of *Sarbanes-Oxley [Sarbanes and Oxley, 2002]* and mandated consideration of fraud in financial statement audits due to *SAS 99* [Jeffery Lehman, 2002], the data shows that occupational frauds are much more likely to be detected by a tip (46%) or by accident (20%) than by audits (28.5%), controls (23.3%) or any other means.
- Implementations of anti-fraud controls appear to have measurable impact on an organizations exposure to fraud.
- Banking ranges as the industry with the second largest median losses (250000 dollars) only exceeded by the manufacturing industry (441000 dollars).
- Occupational fraudsters are generally first-time offenders and work in the accounting departure or upper management.
- Fraud perpetrators often display behavioral traits that serve as indicators of possible illegal behavior (e.g. living beyond their apparent means or experiencing financial difficulties at the time of fraud).

An article in *Bankers Hotline*[Bankersonline, 1992] states that "money laundering gets all the publicity, but there are numerous areas where fraud and theft take place in a financial institution".

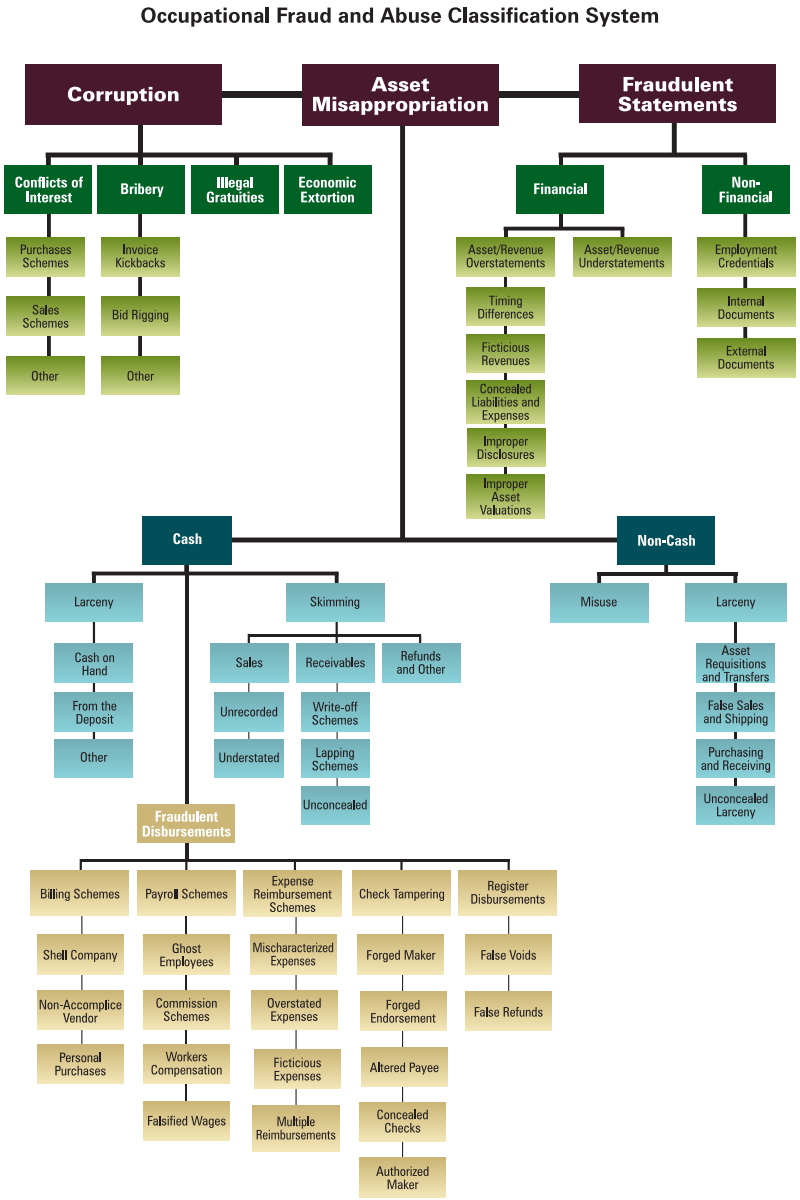


Figure 2.1: Occupational Fraud and Abuse Classification System as proposed in ACFE's 2008 Report to the Nation [Association of Certified Fraud Examiners, 2008]

The authors furthermore state that employees steal because they "truly believe they won't get caught"and "feel overworked and underpaid and are out to "get back" the financial institution". This may gain brisance in times of a financial crisis with reputational challenges and job insecurities. In their initial paper on the current landscape of internal fraud, the *Santa Fe Group* warns that criminal activity occurring inside financial institutions extends far beyond material gain for the individual employee, but may feed activities of fraud rings and other criminal groups around the globe. In their *Forensic Fraud Barometer 2009 [Osborne, 2009]* KPMG reports that "fraud nears record levels in 2008 and worse to come" in the UK with "over £1.1bn of fraud nationwide, the second highest level in 21 years".[3] Summarizing these insights, we can state that even when ignoring reputational losses (which seems to be the case in the reports cited above), internal fraud causes considerable damage. The high level of fraud cases revealed by hints indicates that applied monitoring solutions may have room for improvement. A fraud which is only revealed after the deceived customer complains is a worst case scenario concerning reputation in particular for the banking business, which relies highly on customer trust.

[3] http://www.yhff.co.uk/Fraud

2.1.2 Internal Fraud at Alphafin

We conducted extensive discussions with fraud experts and an analysis of available fraud case reports to gain knowledge about internal fraud cases at Alphafin. As we are not allowed to communicate details of fraud cases due to non disclosure agreements, we present our findings in a summarized form which nevertheless can be used to explain our design decisions in the later parts of this thesis.

Introduction

As mentioned above, a wide variety of internal fraud exists both at the management and employee level. The focus we were given at Alphafin is customer advisor fraud.
A customer advisor manages the accounts of the assigned clients[4]. Client types can range from small private savers to huge corporate customers. As customers are extremely heterogeneous (not only in the size of their assets) customer contact may range from barely existent to good personal relationships. Customer advisors are authorized to trigger transactions concerning their clients assets, under certain conditions without explicit and written customer order. For simplicity, we assign the *Manual Transaction Type (MTT)* to all such transactions. *MTT* transactions are needed to allow for flexibility in customer service. This entails the augmented necessity of appropriate control mechanisms. If the amount of a *MTT* transaction exceeds a defined threshold, operational control measures kick in. Below this threshold, monitoring solutions are needed to handle the large amount of daily processed data.

It is worth noting that while *MTT* is assumed to be more prone to fraudulent activities, a customer advisor may avail himself of other transaction types in combination with counterfeiting required documents and other instruments which are also used by external fraudsters. However, the internal fraudster may have a superior knowledge in comparison to his/her external counterpart. In [Luell, 2005], we listed the premises a client advisor potentially has to commit fraud:

- Detailed knowledge of internal processes and systems (and maybe security flaws)
- Required authorizations to trigger transactions

[4]Customers can have several accounts. Legal bodies in turn may be represented by more than one "customers". As this does not affect our explanations (but is, however, crucial at a more detailed level which is confidential), we treat the terms *customer* and *account* interchangeably in this thesis.

- Knowledge of (usual) behavior and alertness of managed customers
- Fundamental confidence from co-workers and subordinates.

Prominent Elements of Fraud Cases

In the following we list elements which repeatedly occurred in analyzed fraud cases:

Illegitimate speculative trading An important element and often the motivation for further illegal behavior is illegitimate speculative trading. Consider a customer advisor which persuades his clients to approve highly speculative trading transactions promising high profits without sufficiently exposing the connected risks. At the loss of the money the employee tries to minimize the customers financial damage to avoid a customer complaint uncovering the fallible behavior. A more blatant variant is the trading without any customer knowledge, let alone order. The intention is typically to generate profit for personal enrichment without the customer ever learning about the transactions. If the trading does not result in profit, but in loss, the disappearance of assets needs to be covered to avoid exposure.

Extensive asset shifting No matter if money is lost in trading or withdrawn to an unjust beneficiary, numerous transactions are necessary to conceal and obscure the transfers. One purpose for rearranging and shifting money in complex patterns is the temporary balance of missing amounts in the wake of examinations. Another goal is the concealment of the true origin (or destination) of monetary shifts. Doing so, a fraudster will typically act with caution and try to hide such transfers in the mass of legal transactions. The comparatively low amounts of fraudulent transactions (see below) ease the concealment. Similar behavior — but typically at a much bigger scale — is known from the field of money laundering where it is called *Layering* [Altenkirch, 2006].

Exploitation of weakly supervised accounts Not every account is regularly checked by the owner. A fallible customer advisor will know about the accounts which are largely unattended. An attentive customer which regularly receives and checks his/her financial asset documents will soon spot irregularities. More suitable victims do not have a clear overview regarding their assets and barely receive or read documentation. Illustrative examples are highly aged customers

without any affinity for financial affairs and little assistance from their environment. A customer advisor we were allowed to interview stated that a notable number of customers show only minimal interest in their assets. However, we were not able to verify if this is generally true.

Exploitation of diverse vulnerabilities in processes and systems As the information may be highly specific to our industry partner, details are both confidential and of very limited generalizability. It is, however worth stating that, while some of the weaknesses can be eliminated after being revealed, this is not true for all. Some soft spots may be intrinsically tied to process efficiency and flexibility. Processes and systems cannot be always optimized for maximal security.

Gradual increase of fraudulent transactions Fraudsters will typically start with very cautious, scarce transactions and gradually intensify the activity as the confidence not to be caught increases over time. It is tempting to try again what worked once.

Single perpetrators, no organized crime While it may be attractive for organized crime to infiltrate companies with the goal to steal data, doing the same to steal money may not be the best choice (at least in the situation at hand). The (expected) level of supervision, identifiability and non-repudiability is too high. As stated before, we are concerned with opportunity crime and acts of desperation after minor or major offenses. We didn't meet a single case where a collaboration of several fraudulent employees was observed or an employee was involved in organized crime. Experts repeatedly affirmed that organized external or internal accomplices are highly unlikely. However, it is imaginable that a fraudster incorporates unsuspecting subordinates or customers to a certain level.

Relatively static, recurrent fraudulent behavior As a consequence of low professionalism, internal fraud as we met it is typically less sophisticated than elaborate external fraud or anti-money laundering methods where fraudsters quickly adapt to new situations and circumvent new detection systems. As perpetrators are not able, nor trying to learn from each other, observed fraudulent behavior is relatively static over time, apart of course from subsequently closed flaws.

Extremely rare occurrence We found very rare occurrence of internal fraud. Of course, this observation is solely based on revealed cases. A rough estimate of fraudulent transactions in the

given data set with a time window of the two most recent years lies in the range of $1 \times 10^{-7}\%$ to $5 \times 10^{-7}\%$ of all transactions[5]. This value is considerably lower than the ones reported in [Phua et al., 2005], giving an overview of fraud vs. legal ratios in the data sets used in related work. While most of the data sets exhibit fraud proportions between 10% and 25%, the lowest ratio declared is 0.1% in [Kim et al., 2003b; Cahill et al., 2002].

No generalizable delinquent profile Fraud experts at Alphafin reported that an extensive project was conducted with the goal to describe "the typical fraudster"in terms of age, duration of employment and so on in order to identify periled employees. The findings were that a profile could not be defined due to the small number of examples and the considerable heterogeneity which was much higher than expected.

Low transaction amounts Basically, additional control mechanisms kick in if a defined threshold is exceeded which complicates covering. Beyond this, fraudsters are well aware that the higher transactions are, the higher is the risk that they cause a stir. So amounts of fraudulent transactions are kept as low as possible. Needless to say, on the other hand the expected profit has to be worth the risk so trifle amounts will also not occur in fraudulent transactions.

Recurring patterns (Chain Transactions, Smurfing) This characteristic is closely connected to the extensive asset shifting. As these patterns are central for the design of our approach, we will discuss them in greater detail in the following section.

On Recurring Fraud Patterns

In the visual analysis of known cases we encountered a number of patterns which were confirmed by fraud experts to recurrently serve as indicators for fraudulent behavior in manual investigation. Those patterns can be summarized under the terms *Transaction Chains* and *Smurfing* activities. Transaction Chains and Smurfing are well-known instruments for concealing actions in a wide variety of illegal activities.

[5] only accounting for internal fraud

We define a *Transaction Chain* as follows:

Definition 1 *A transaction chain is the transfer of money from a source to a target via 1-n intermediary points (e.g. accounts) within a short time period (typically one to a few days) where the amount of money leaving the source approximately equals the amount of money arriving at the target.*

Each of the three roles (source, intermediary, or target) has its unique characteristics and implications for suitable accounts taking this role. The detour over one or more intermediary nodes is typically necessary to avoid the obvious transfer from a basically suspicious constellation of source to target accounts. Money is typically only kept for a short time at intermediary accounts because they are no safe harbors. Chains can also emerge from efforts to balance discrepancies on victim accounts on a short-term basis. In this case, a loop chain where the source and target node are identical is also possible. *Structuring* or *Smurfing* denotes the splitting of money transfers into numerous smaller transactions for further concealment. Figure 2.2 shows examples of different complexity. In its most simple form, a chain consists of single, approximately equal transactions (a). In a more elaborate version, the transferred money is broken into several smaller transactions from one station to the other (b). Another possibility is the distribution of the amount to several intermediaries (and/or targets) (c) . While complex pattern incorporating structuring may seem "more promising"for a fraudster at first sight, this technique requires more triggered transactions and is more laborious. A fraudster may want to avoid this because he/she feels (i) safer initiating as few transactions as possible as additional transactions increase the possibility of random discovery, or (ii) so safe that he/she avoids the additional work and delay in setting up elaborate Structuring. Interestingly, even though these patterns may be well known they seem to play a relevant role in internal fraud.

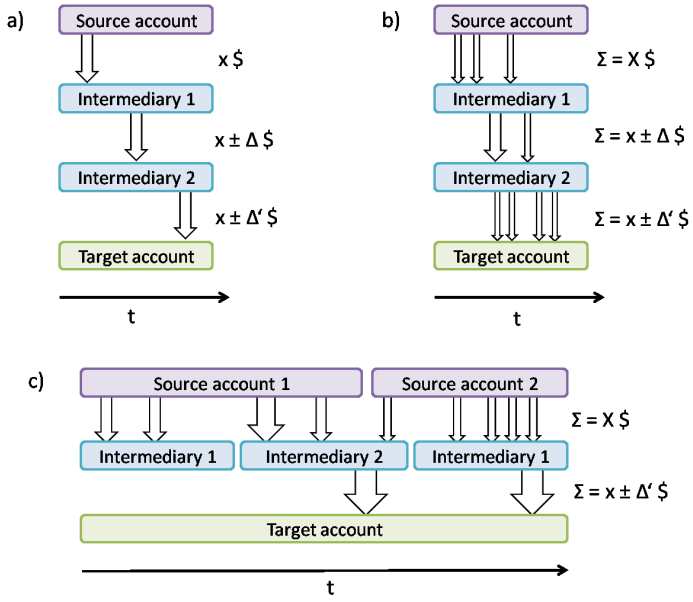


Figure 2.2: *Transaction Chains* and *Smurfing*

Why the Focus on Transaction Chains and Smurfing?

While our approach allows to define and retrieve a wide variety of patterns where the relation of attribute values between nodes and/or edges in the graph[6] are of special interest, we focus on Transaction Chains and Smurfing patterns. The reasons for this decision are the following:

- These patterns have repeatedly been observed in connection with fraud cases. However, none of these cases has been revealed on the basis of these patterns as adequate monitors do not exist.
- The assumption seems reasonable that these patterns occur independently from the exploitation of specific flaws. They may therefore have a higher general discriminative power than vulnerability specific patterns as the according weaknesses may be closed by operational measures.
- Although chains and smurfing are denoted as typical of fraud by experts, we encountered

[6] Accounts (or customers) and transactions between them can be interpreted as a graph, where accounts represent nodes and transactions represent edges in the graph. This perspective is common for structural analysis, for example in Social Network Analysis [Wasserman and Faust, 1994].

comparatively low knowledge about their discriminative power. How common these patterns are in legal behavior and under what circumstances they may occur is largely unknown. To evaluate this and provide the knowledge to human experts has become one of the goals of this thesis. This implies that we do not solely want to find fraud, but find patterns of interest and learn about their general occurrence, which helps to refine the experts model of fraud.

- It has never been our goal to develop an all-in-one-solution but to complement applied approaches. We tried to classify analyzed cases roughly into three categories.
  - Cases in the first category require information beyond the available data, (e.g. personal relationships, general financial status) for detection.
  - The second category contains cases that can be detected with technically straightforward threshold filters over the stream of transactions (for example implemented in *SQL*).
  - Fraudsters in the last group more or less successfully conceal their activities by combining groups of inconspicuous transactions that disappear in the mass of similar legal movements. The use of *Chain Transactions* and *Smurfing* was intense in this group.

For the above reasons, it is the members of the last group that we decided to focus on. It has to be stated that we do not argue or expect that almost all internal fraud will contain transaction chains and structuring. But we argue that it is worth examining those structures as they are helpful means to circumvent existing fraud detection solutions.

2.1.3 Compliance

Compliance can be defined as "acting according to certain accepted standards" . Compliance violations according to company regulations may occur due to (possibly willful) ignorance or negligence. Customer advisors may not follow mandatory processes, make use of a system in other than the intended way or act uneconomically. Internal fraud can be seen as severe non-compliant behavior with criminal intention. The detection of non-compliant behavior is therefore related to internal fraud detection.
Without delving into the topic it turned out that our approach may have considerable potential in this area (see section 4.3).

2.1.4 Money Laundering

One of the most prominent forms of fraud is money laundering. Countermeasures, or anti money laundering (*AML*) is of particular but not exclusive interest for the financial industry. Legal regulations commit financial and non-financial institutions to identify and report suspicious money transactions to financial intelligence units in most countries [Moll, 2009]. Legal regulations are for example defined by the Bank Secrecy Act of 1970 and the USA PATRIOT Act in the U.S. and by the Swiss Anti Money Laundering Act of 1998 (*SAMLA*). Since 2009, the Swiss Financial Market Supervisory Authority (*FINMA*) is in charge of the governmental supervision in Switzerland. The Financial Action Taks Force (*FATF*) is an inter-governmental organization founded in 1989 by the G7 forum. The *FATF* currently comprises 32 member jurisdictions and 2 regional organizations, representing most major financial centers in all parts of the globe [FATF, 2009].
Due to organizational changes during the course of this thesis, the focus on internal fraud detection was switched to certain aspects of *AML*, in particular the *Know Your Customer* (*KYC*) principle. The approach developed for internal fraud detection was therefore extended and adapted to evaluate its potential in the area of *AML* as well (see section 4.4.1). However, in contrast to internal fraud, we did not have the possibility to analyze the characteristics of money laundering in Alphafin extensively, but relied on *AML* expert input. Therefore we refer to the numerous publications available in this domain, e.g. [Altenkirch, 2006].

2.1.5 Other Types of External Fraud

The issue of external fraud was not broached in this thesis—with the exception of a quick glance at money laundering. Potential of the proposed solutions in detecting external fraud in Alphafin remains future work.

2.1.6 Countermeasures

Countermeasures can have two basically different goals:

- Fraud prevention
- Fraud detection

Avoiding fraud is preferable to detecting it when it already happened, but is not always possible. Safeguarding and keeping business processes flexible and efficient may antagonize each other. Preventive countermeasures can be implemented at two different levels:

- Organizational
  Qualified structures and competence orders can avoid or complicate illicit acts and close organizational flaws. Training and company culture may be used to raise fraud awareness.
- Operational
  measures adapt business processes and applications to hinder violations and misuse. Extensive control measures come at the risk of upsetting employees and to cumber daily business. An example is the two-person integrity for sensitive processes.

Detection requires analytical countermeasures. In [Luell, 2005], we distinguished four possible procedures for analytical fraud detection:

- Day-to-day inspection of the total data by fraud experts
  All the data is searched for conspicuous patterns which are investigated in detail. Due to the high manpower requirements, this approach is only feasible for a very limited amount of data and if fraud has to be detected at any cost.
- Samples are drawn from the total data and inspected.
  The sample choice may be influenced by expert know-how to a certain extent, but remains

a more or less random process. Sample size might be chosen on the basis of disposable manpower. This approach is a possible choice if investigation of the total data is infeasible. Its relatively undirected nature calls for more effective approaches. In particular if fraud is extremely rare, the chance of a "lucky strike" is far too low.

- Based on concrete hints and complaints an explicit, a constricted subset of the data is investigated. A customer may notice discrepancies in asset documentation and informs the company. A case-based investigation is the consequence. Case-based detection might be a widespread and accurate approach, but it is, basically, "locking the stable door after the horse has bolted". At this point, at least reputational loss has already happened, and the goal is damage limitation. Detecting fraud before the cheated customer realizes it is definitely preferable.

- The total data is automatically searched for potentially critical subsets which are presented to human experts for closer investigation. Depending on the accuracy of the identification and retrieval of critical subsets, this approach may allow for a significant improvement of efficiency and effectiveness in comparison to the procedures mentioned above. Different strategies for identifying adequate concepts and retrieving patterns of interest are discussed in the section 2.3.

As expected, our approach falls into the last category of possible procedures.

2.2 Fundamentals of Data Mining

What is data mining[7]? This term denotes the computer-aided discovery of patterns of interest in data bases and therewith revelation of valuable information previously hidden in the potentially massive amount of data. Consider the data of purchases in a supermarket. Hundred thousands of shopping baskets may be recorded every day for a large supermarket chain. Data mining algorithms may search the data and find patterns like "*Customers which buy red wine and French cheese will, with very high probability, also buy baguette*" or "*In rural regions, female customers at the age of 40 to*

[7] This section is intended for the reader not familiar with data mining and related topics. Others can safely skip it. The goal is to provide a very short high level description of the concepts required to understand the assessment of related work and the derivation of our approach decisions.

50 can be classified as high-value customers."Data mining is the core of the *KDD*-Process (Knowledge Discovery in Databases), which is concerned with the selection, preprocessing, transformation and mining of data, including the evaluation and interpretation of the results. Data mining is just a tool, human (business) experts evaluating and considering the value of identified patterns is typically indispensable. Data mining algorithms generally aim for prediction or description. Prediction makes use of some attributes in the database to predict unknown or future values of other variables of interest. Description focuses on finding human-interpretable patterns describing the data [Fayyad et al., 1996]. At a slightly more detailed level, data mining can be divided into the following tasks (description is based on [Fayyad et al., 1996]):

- Classification is a learning function that maps (classifies) a data item into one of several predefined classes (e.g. the classification of a customer into "defrauded"or "not defrauded")
- Regression is a learning function that maps a data item to a real-valued prediction variable (e.g. the prediction of a customers lifecycle value in dollars)
- Clustering is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data (e.g. identifying relevant target groups of customers)
- Dependency modeling focuses on describing dependencies and associations between data items (e.g. finding products which are commonly purchased altogether)
- Change and deviation detection focuses on discovering the most significant changes in the data from previously measured or normative values (e.g. finding phones with uncommon usage patterns for fraud detection)

Another crucial concept is the distinction between *supervised* and *unsupervised* methods. Supervised methods learn predictive models from training examples. Training examples belong to one of the classes used later for prediction. The attribute denoting class membership is called target variable or label. For example, if a sufficient amount of accounts known to be defrauded is available (and not defrauded accounts likewise) a model can be calculated which finds significant differences (based on the available attributes) in the two classes. The model is then applied to future data, predicting class membership (if a account may be defrauded or not). Supervised methods require the availability of target variables (labels), which is not always given in a real world problem. Classification and regression are typical exponents of supervised methods.

Unsupervised methods do not require (class membership) labels and focus on a compact description of the data items. In comparison to supervised methods, they often have a more explorative character. A particular significance relating to fraud detection has *deviation or outlier detection*, based on the assumption that uncommon behavior is a good indicator for fraudulent behavior. Other methods try to identify common behavior or behavior that slightly deviates from common one.

2.3 Related Work

A considerable amount of research work has been published in the field of data-mining-assisted fraud detection. Existing survey papers give an overview of publications [Maes et al., 2000; Phua et al., 2005; Kou et al., 2004; Bolton and Hand, 2002]. However, a systematic analysis with reference to crucial differences of the approaches and their premises to be effective has never been conducted.

In the following, we organize existing work based on the according *prior knowledge level* as it provides a crucial differentiator. First, the three levels of prior knowledge are defined and described. After this, research in all three categories is described.

Concluding, we consider the field of visualization and visual data mining in a nutshell. Complementing explicit fraud detection papers, we include some research from other areas that is relevant for the project at hand.

Fraud detection is a wide area. Specific detection problems can differ from each other in numerous aspects, which, in turn, leads to a wide variety of adequate solutions. After giving an overview of the existing research, we identify those aspects and provide a framework to support assessments of future fraud detection problems and possible approaches based on related work. Furthermore, we will motivate the design decisions for our approach by applying this framework to the situation we met at Alphafin.

2.3.1 Classifying Related Work: Prior Knowledge Levels

When considering different approaches for application to a real world problem, the available prior knowledge on fraud is a crucial factor. While in one case, fraud is known to be a problem but otherwise a largely unknown concept, there may be extensive knowledge of revealed fraud in another case. Basically, three levels can be identified:

No or very low prior knowledge If a business is considered under the aspect of fraud prevention or detection only for a short time, knowledge based on experience is lacking. The fact that fraud detection is considered at all suggests however that a certain amount of fraud cases has been discovered and minimal knowledge is available. Only one paper mentions that the in-

troduced data-mining-assisted system constitutes the initial countermeasure and therefore prior knowledge is very limited [Major and Riedinger, 2002]. Such a system has completely different premises than a solution which can be based upon extensive expertise. A common approach to this problem is not to have a system identify "fraud" (as this is a widely unknown concept for the time being) but to search for "uncommon behavior" which, of course, has to be defined in an appropriate way. The identified behavior is subsequently delivered to human experts for closer investigation and evaluation. This approach assumes that there is an adequate concept of "uncommon behavior" which is able to discriminate between fraudulent and normal behavior. Such a system requires extensive efforts by a human expert, which learns the concept "fraud"while working with the system. Ideally, the expert may feed the gained knowledge back to the system to gradually improve its discriminative power. Typically, the number of "false positives"(false alarms) is very high at the beginning and reduces as the human and the system refine their knowledge about the mechanisms of fraud.

Existing identification knowledge ("labels") We define identification knowledge as follows: *Identification knowledge* is knowing if an instance is fraudulent or not. In this case, the data is typically "labelled"[8] For example, phone fraud experts may know which of the calls logged in the past are fraudulent. They may, however, not have an extensive knowledge about the mechanisms and properties of the fraud cases, what we call model knowledge (see below). This situation is ideal for an approach which Fawcett and colleagues describe as "discriminating method" [Fawcett et al., 1997]: based on known fraudulent examples a model is calculated which describes the discriminating factors between fraudulent and normal behavior. This approach can also be valuable in the presence of prior (human) knowledge about fraud mechanisms. Being able to consider more data than a human expert, calculated models can display previously unknown interrelations or can be used to support informal hypotheses. As the presence of a sufficient number of identified fraud cases (used for model calculation) typically implies a certain prior model knowledge which lead to the discovery of those cases, this approach is assumed to offer model knowledge refinements rather than completely new knowledge. In other words, the models are, naturally, only calculated on the basis of previously identified examples and therefore will not have the ability to find completely new fraud tactics. Completely new, innovative fraud brings

[8] E.g. there is an attribute providing the information if an instance is fraudulent or not.

the human experts back to the "no or very low prior knowledge" situation and the appropriate methods mentioned above.

Another crucial consideration when using calculated models is the interpretability. The active substantiation of a suspicion (for example by interrogating the suspect) is only possible if the investigator is able to reconstruct the reason for the initial suspicion. No suspect will be reported without a human expert knowing why (but only based on a score based on a black-box-model). If the calculated model is human-interpretable (e.g. a decision tree), suspicion validation can be accomplished directly. If the model is not human interpretable (e.g. a neural network), its output has to be validated by a human expert e.g. directly based on the data, where it is, to some extent, replaced by a human model. Some research work appears to completely disregard this aspect.

Existing model knowledge This type of prior knowledge is more elaborate than identification knowledge. Fraud experts may know exactly what they are looking for from experience, but they lack the appropriate tools for an efficient and effective search. In contrast to the approach mentioned above, the models are not calculated by an algorithm, but defined by human experts. These models can, e.g., be intentionally fuzzy to allow for an extension of knowledge about fraud when evaluating the results ("seeing the borders between fraud and normal behavior"), which again points back to the "low prior knowledge setting". An advantage of this approach is that the model is not constrained by the model calculation method, but only by the richer expressivity of the human brain and the mechanisms built to retrieve the patterns defined by the model. A disadvantage is the possible discrepancy in the identification knowledge and the model knowledge of the human expert (e.g. because the model is too complex to be captured by a human expert — or the amount of data is too big to consider exhaustively). Prior model knowledge motivates the use of pattern matchers to retrieve interesting data.

Another relevant categorization can be made concerning the available format of prior knowledge:

- Machine readable

  Machine readable identification knowledge corresponds to “labels”or “targets” in Data Mining, which is a requirement for the model calculation with supervised Data Mining algorithms. Machine readable model knowledge is a typical result of Data Mining and other knowledge engineering techniques.

- Informal, not machine readable

  Informal identification knowledge requires manual effort. The transfer to a machine readable format suggests itself, but is not always possible (as in our case). Informal model knowledge is the typical example for human model construction, e.g. if model calculation is infeasible or not considered valuable.

2.3.2 Fraud Detection Based on Very Low Prior Knowledge

The detection of abusive use of a retail transaction processing system by employees is the goal in [Kim et al., 2003b]. It is therewith one of — to our knowledge — only two academic publications concerning analytic internal fraud detection at the employee level. The authors argue that as the retail sector often does not possess sufficient expertise about potential and actual frauds, an anomaly detection approach to fraud has to be employed. The idea is to find temporal association rules between transactions which are uncommon but still not extremely rare. These association rules are then used to create detectors. Experts should then be able to decide which detectors are valuable. Valuable detectors are then cloned and mutated to broaden the scope and make the search more "fuzzy" . This approach is inspired by the human immune system, using positive and negative selection [Kim et al., 2003b]. Although the vision was to build a ready-to-use product, the project turned out to be too ambitious and stopped at the prototype level as mentioned in the final report [Kim et al., 2003a]. However, extensive expert evaluation resources seemed to be available, which is desirable in the case of an anomaly detection approach. Evaluation results were not as expected as the detected and fully investigated anomalies turned out to be legal behavior.

A solution which is occasionally used in anti-money-laundering is the *Peer Group Analysis* [Weston et al., 2008]. The idea is to define a peer group which contains the objects most similar to the target object. Peer groups can for instance be defined based on clustering algorithms or based on business knowledge. The objects and peer group summaries (which describe the "average" behavior within a peer group) are monitored over time. If an object starts to exhibit behavior which is substantially different from the average behavior of its peer group, it is considered suspicious. Changes which affect the whole peer group (e.g. due to changes in the market situation) are masked, which avoids the generation of false alerts. The authors argue that the distinguishing feature of the peer group analysis lies in its focus on local patterns rather than global models. The unusualness of an object is not measured on the basis of the whole population but on "similar"objects. The choice of an adequate similarity definition is crucial for this method. This approach makes prominent assumptions, which are not mentioned explicitly in the publication. In particular, the authors assume that behavior change is a valuable indicator for fraudulent behavior, which may not always be true. We will discuss this in greater detail when introducing

our unifying framework in section 2.4. In a recent publication [Weston et al., 2008], an in-depth discussion of credit card fraud detection using peer group analysis is provided.

A similar solution for the problem of low prior knowledge[9] in a topic where fraud detection has not been done before is proposed in [Major and Riedinger, 2002]. This approach makes use of an "operations cycle" and a "development cycle" to detect fraud in health care claims. First, a Peer Group Analysis variant is used to find health care providers which "stand out from the mainstream", which are then presented to a security unit. In the development cycle, rules should be induced based on the expert analysis of the outliers. As in [Kim et al., 2003b], these rules are proposed to be cloned and mutated. Details about the development cycle process are not given. This system is supposed to "address a class of identification problems that are more likely to be encountered in business than in science or engineering" and is therefore mostly application-oriented. The question arises if outlier detection is a sensible discriminator for fraud identification. We will get back to this issue when we discuss applicability issues below.

An alternative approach to the problem of missing identification knowledge is proposed in [Brockett et al., 2002]. It is not based on outlier detection as most of the related work, but tries to achieve a classification (or at least a fraud suspiciousness ranking) without given class labels. Instead, experts are required to look at each attribute used for prediction and rank the corresponding attribute values in terms of likelihood of suspicion. Attributes are assumed to be ordinal. Based on this ranking, the attribute values are transferred to numerical RIDIT-Scores[10] [Brockett et al., 2002]. In contrast to the obvious solution of just assigning integers - as for example, the rank - to the possible attribute values, RIDIT-Scores do not presuppose equal interval spacing and, in addition, are able to reflect "abnormality"of an attribute value (which reflects the concept of entropy used e.g. in decision tree algorithms). Summation of the RIDIT-Scores of each attribute values of a given instance leads to an overall score which can be used for classification or ranking. Apparently, this approach requires a certain amount of model knowledge from experts to do the attribute value ranking. It could be argued that it therefore falls into the category of approaches with given model knowledge.

[9] The authors Major and Riedinger use the term "fragmentary, microlevel knowledge".

[10] This term was introduced by [Bross, 1958] and denotes a scoring model for ranking attribute values according to an underlying latent variable (in this case fraud likelihood). In contrast to the obvious solution of just assigning integers - as for example, the rank - to the possible attribute values, RIDIT-Scores do not presuppose equal interval spacing and, in addition, are able to reflect "abnormality"of an attribute value (which reflects the concept of entropy used e.g. in decision tree algorithms)

An approach based on a first-order markov chain for phone fraud detection is discussed in [Hollmén and Tresp, 1998]. The generative model used exists of two hidden binary variables, representing if an account is currently victimized by a fraudster and if the fraudster in question currently performs fraud, respectively. The observed binary variable is representing if a mobile phone is currently being used. The experiment was started in an entirely unsupervised experiment, but due to poor performance, available information which accounts were victimized by fraud was used for parameter estimation. Further examples which make use of outlier detection and behavior change analysis are [Yamanishi et al., 2000] and [Burge and Shawe-Taylor, 2001].
In [Xu et al., 2006], a method for monitoring behavior to detect online attacks is proposed. As in many other approaches, a behavior profiling is followed by a behavior monitoring. This is done on both the individual and the system level. The authors argue that system level monitoring may help to detect system flaws which are exploited by many users (for example, obtaining game points without playing in an online game), but may not be detected at the individual level as the impact is too low.

2.3.3 Fraud Detection Based on Prior Identification Knowledge

The situation of available identification knowledge maps directly to the use of supervised data mining algorithms, which require the corresponding labels. As expected, most of the related work meeting this initial situation makes used of supervised algorithms. In the field of management fraud, [Fanning and Cogger, 1998] makes use of a Neural Network (*AutoNet*) to learn a model based on constructed training data. Informal identification knowledge in the form of *SEC Enforcement Releases*[11] is converted to labels for training. Entity attributes are company and management structure descriptions —this is one of very few examples in fraud detection where the focus does not lie on events but objects.

A considerable amount of research in the fraud detection area is based on ready-to-use data with labeled examples. The focus lies on the design of the data-mining method, application issues are completely ignored. In this setting, evaluation is straightforward. A common approach is the use of multiple classification models, which are combined using meta-learning[12]. Phua [Phua et al., 2004] uses this approach to detect illegitimate car insurance claims. An explicit cost model which considers average costs of investigation and average cost per claim is used for evaluation. The performance is measured in cost savings, which, of course, is attractive but may be infeasible for most real world evaluations as the actual costs may be very hard to determine. Meta-learning is also proposed to attack the problem of massive data, highly skewed data, and variable classification costs [Chan et al., 1999; Stolfo et al., 1997; Stolfo et al., 1998]. The idea of combining local fraud classifiers, which are calculated in different financial institutions into a global detector using meta-learning, is introduced in [Prodromidis and Stolfo, 1999]. This should allow companies to share knowledge about fraud without exchanging sensitive data and allow for a global detector which "will be able to pick up patterns of fraud that are not detectable at the local level [...]". The idea seems attractive as, e.g., in money laundering detection, the limited local views of single financial institutions form a crucial limitation. However, it remains unclear if models which are unspecific enough to be interchangeable without giving away sensitive data can be combined to a more expressive meta-classifier. In particular, without revealing customer information, activities of a person in different financial institutions cannot be mapped to each other. Another issue are

[11] U.S.Securities and Exchange Commission Enforcement Releases, http://www.sec.gov/divisions/enforce/friactions.shtml

[12] Meta-learning denotes the learning on the basis of results from previously applied learners. An extensive survey can be found in [Vilalta et al., 2004].

differences in schema definitions of the databases, which lead to incompatible classifiers. [Prodromidis and Stolfo, 1999] describes a technique called "bridging"to overcome this problem, which basically is a combination of well-known data preprocessing steps. This approach is reported to be successful in an experimental setup using credit card fraud data across two participating banks.
[Maes et al., 1993] is an example of straightforward application of existing data mining algorithms to an "ideal" data set: it uses Bayesian and neural networks for credit card fraud detection data. Unfortunately, details about the used features are not given. Bayesian Networks Models are used in [Ezawa and Norton, 1995] to predict uncollectible debt.

No success using standard machine learning techniques is reported in [Fawcett et al., 1997] and [Fawcett and Provost, 1996] for phone cloning fraud detection. Two problems are identified: First, a call that is unusual for one customer can be typical for another customer. Context information to account for this fact is not directly available in this case, so it is derived from historical data specific to each account. This leads to the detection of changes in behavior rather than absolute indicators of fraud. Second, fraud identification on the basis of individual calls is assumed to produce an unacceptable true positive/false positive ratio. The single event perspective is therefore switched to an aggregated event or single object perspective, "smoothing out the variation and watching for coarser-grained changes that have better predictive power". The proposed approach generates fraud rules based on each separate account using a beam search. In a rule selection step, the most general rules are selected. These general rules are then turned into account specific profiling monitors, that is, the rules are customized to each account. If a monitor registers deviation from the modeled normal behavior, it will generate an alert. In a final step, evidence from the monitors is then combined to an overall score. Evidence combination weights and the threshold above which an alarm should be issued is learned using a standard learning algorithm. This approach is similar to the peer group analysis [Weston et al., 2008] in two aspects: it also makes use of local instead of global models and searches for behavior changes.
Cahill et al [Cahill et al., 2002] extend this approach by changing the focus from a time-driven (account summaries on a daily basis) to a event-driven model, weighting recent calls more heavily and by using adaptive, updatable user profiles they call "account signatures". Account signatures are based on estimations for multivariate probability distributions, describing which call features

are likely for the account, and which are not. The authors state that "fraud typically results in unusual account activity", this probability is the "right background to judge fraud against". In addition to the account signature, a "fraud signature" is used, which requires labeled data.

A more general view on the field of activity monitoring and corresponding evaluation issues is given in [Fawcett and Provost, 1999].

A completely different approach is used in [Burge and Shawe-Taylor, 2001]. An expert system using simple fuzzy logic based rules is proposed. The fuzzy rules account for subjective nature and therefore with the ambiguity of the parameters. The system is designed for insurance claim detection but is only tested on hypothetical data.

The recent survey of data mining-based financial fraud detection research [Yue et al., 2007] reports that the large part of publications relies on supervised methods as regression and neural networks. This is not surprising as the use of available labeled data is very convenient for readily generating research results and publications.

2.3.4 *Excursus*: Relation Based Approaches

So far we have seen that the large part of analytical fraud detection research is based on the analysis of single events or objects. This seems to be true for very recent publications as well, although fundamentals of relation based mining and pattern matching topics are actively researched today. The reason for this may lie in the nature of available data for fraud detection research. For instance, when analysing phone calls or credit card transactions, a large part of the relevant network may lie beyond company borders and is therefore not available[13]. However, methods concerning object or event relations seem to be more present in other adversarial domains as anti-money laundering or terrorist detection [Phua et al., 2005]. Of course, the application of relation based methods requires the data to be of relational nature, that is, consist of events (as phone calls or transactions) and objects (as accounts or mobile phones). Most data used in related work satisfies this precondition. Typcial exceptions are financial statement or insurance fraud.
Consequently, the related work discussed in this section is not limited to fraud detection research, but relation based mining and pattern matching in general. Mining relationship network data which typically can be represented as a graph is an relatively new research topic. Numerous terms and concepts recently emerged which are often not easy to differentiate. One survey paper [Washio and Motoda, 2003] for example proposes that "graph data mining is more geometry oriented and relational data mining is more logic and relation oriented", while another [Chakrabarti and Faloutsos, 2006] states that "relational learning typically focuses on finding small structures/-patterns at the local level, while graph mining looks far more at the global structure". A great deal of similarities exist between the work in link analysis, hypertext and web mining, relational learning, inductive logic programming and graph mining. The excellent survey by L. Getoor and C. Diehl [Getoor and Diehl, 2005] introduces the term *link mining* as the intersection of these research areas and identifies eight link mining tasks in respect to their focus on objects, links or whole graphs (See table 2.2). We will stick with this terms when discussing example approaches.

One of the most prominent tasks in link mining is the subgraph discovery problem of finding frequent patterns in graphs. The idea is easily understandable: given a graph, find common substructures which repeatedly occur on a local level. An intuitive application is the identification

[13] We met the same problem in a cross-selling analysis performed for a telecommunication company [Weiss, 2006], where relation based approaches turned out to be inappropriate.

Object Related Tasks	Link-Based Object Ranking
	Link-Based Object Classification
	Object Clustering (Group Detection)
	Object Identification (Entity Resolution)
Link Related Tasks	Link Prediction
Graph Related Tasks	Subgraph Discovery
	Graph Classification
	Generative Models for Graphs

Table 2.2: Link Mining tasks

of common atom/bond structures in molecules. *Subdue* was one of the first frameworks tackling this problem and has been constantly extended [Holder et al., 1994; Cook and Holder, 2000; Ketkar, 2005]. It identifies the subgraphs which best compress a graph in terms of the *minimum description length principle*[14] and replaces the corresponding instances with a placeholder node, hereby compressing the graph. The algorithm is then applied again to the compressed graph iteratively which allows for finding common subgraphs consisting of previously identified common patterns. Iterative application on the increasingly compressed input graph can also be used for graph-based hierarchical clustering [Jonyer et al., 2000], resulting in a clustering tree of nested substructures. Domain knowledge can be used to influence the scoring of the patterns for routing the search process to a certain extent. A match cost function can be used to bring in inexact matching.

The counterpart of the common substructure finding task is the mining for structural anomalies, which has been recently done in *Subdue* [Eberle and Holder, 2007]. Anomalies are seen as modifications, insertions or deletions on common substructures. For this purpose, the algorithm searches for structures which are similar to common substructures, however dissimilar enough to appear to be (or more precisely, to contain) an anomaly in the above sense. The authors state that their definition of an anomaly is different from most approaches which are looking for unusual or "bad"patterns. As e.g. money launderers try to imitate legal transaction patterns, this approach could have potential in anti-money laundering and possibly in fraud detection, which is contemplated, but not evaluated in the paper . However, the introduced approach presumes a number of eminent assumptions, e.g. that the majority of the input graph consists of a common pattern and anomalies do not deviate more than a fixed threshold from this pattern. Furthermore, the

[14] for extensive information on the minimum description length principle, see e.g. http://www.mdl-research.org/

graph is assumed to be "regular" however, it remains unclear what the authors exact definition of a regular graph is.

In the second one of the two internal fraud detection papers at employee level known to us [Eberle and Holder, 2009], the authors use graph based anomaly detection for the detection of insider threats in business transactions and processes. Anomalous instances of structural patterns are discovered in data that represent entities, relationships and actions. The authors assume again that the majority of the analyzed graph consists of a normative pattern and only limited modifications are done in the case of an anomaly that is, anomalies are similar to the normative pattern.

Supervised graph classification is also done with the aid of *Subdue* [Holder et al., 1994]: Given a training set of positive and negative graphs[15], structures are searched which are common in the positive graphs, but not common in the negative graphs, therefore building discriminative structures between the two classes.

A stronger focus on context-induced interestingness for frequent pattern mining is proposed in [Berendt, 2005]. The author introduces a taxonomy on the graph by simply adding more general terms to the nodes of the graph (e.g. "animal"instead of "bird"). Frequent patterns are then mined using the general terms and called "abstract patterns". However, the algorithm not just returns the abstract patterns, but also the "individual patterns"constituting them (on the "bird level"). It therefore delivery "abstract pattern-frequent individual patterns"which are supposed to be of interest.

Another publication [Zaki et al., 2004] proposes a whole system for frequent pattern mining for massive data sets including a storage and persistency manager. This is a good example for approaches which may have a very good scalability and performance but are infeasible for research under constrained real world conditions as the installation of whole new systems is often not allowed. [Washio and Motoda, 2003] gives an extensive overview on other frequent pattern mining approaches, categorizing them into greedy search based, ILP[16] based, inductive DB based, graph theory based and support vector based methods. In crime related topics, link based object ranking and classification is more common than frequent pattern mining. A prominent example is the suspicion scoring based on *guilt-by-association* as described in [Macskassy and Provost, 2005]. Suspicion scores of nodes in the network are estimated by counting malicious associates. As the

[15] In this case, whole graph are labeled as positive or negative.
[16] Inductive Logic Programming

scores of the individual nodes influence each other (i.e., their neighbors), the scoring algorithm is not "finished"after the first pass, but can be called iteratively on the node scores changed in the previous pass. Scores are propagated through the network until all suspicion scores stabilize. Convergence is enforced by using a *simulated annealing* to avoid oscillation between different end-states. This approach is based on the relaxation labeling algorithm introduced in [Chakrabarti et al., 1998], where it is used to categorize hypertext using link information. In their paper introducing *communities of interest* [Cortes et al., 2001], the authors point out the problem of record linkage. For example, if a fraudulent account is detected and closed, it is possible that the fraudster will use another account under a different assumed identity. The goal is to find subgraphs with a significant overlap to the behavior of the previously identified "fraudulent subgraph"to reveal the fraudster behind the assumed identity.

Another straightforward method for link-based object classification is described in [Bernstein et al., 2003; Bernstein et al., 2002]. The approach is based on two different types of vectors: *Entity vectors* represent entities in respect to their strengths of relationship to some defined background entities (e.g., nodes with known labels). *Class vectors* represent classes in respect to the strength of relationship between the class and the background entities. Entity vectors and class vectors can be combined for building a scoring which is used for classification of ranking. As in guilt-by-association [Macskassy and Provost, 2005], this approach assumes that the interlinkage between class members is strong enough to build a good classifier solely based on linkage information.

Another link analysis approach which may be of special interest for fraud and crime detection are *center-piece subgraphs* [Tong and Faloutsos, 2006]. The center-piece subgraph problem is based on a set of query nodes Q, which are generally speaking nodes of special interest. The task is to find a subgraph with strong connections to all or most of the nodes in Q. This is done by using a random walk strategy starting a the query nodes, identifying good "connection" nodes between query nodes. The approach exhibits similarities to *TMatch*, which is one of the approaches we propose for the detection component. The main difference is that the center-piece subgraphs approach is more focused on identifying "centerpieces" (central nodes) while TMatch also does identify central nodes, but is more focused on the structure and relations between all nodes in a subgraph of interest. Most work in link-mining is focused on static graphs and completely ignores the time dimension. However, in many real world graphs, weights between nodes are often not

static but changing with time, forming a time series [Jin et al., 2007]. One possibility to account for changes over time is to identify trends in time series of graph elements and then mining for common substructures in the graph relating to trends. The authors of the *Trend Motif* approach [Jin et al., 2007] assume given weights for vertices which change over time. Maximal consistent (increasing/ decreasing) trend intervals are identified. During an identified trend interval, the corresponding node is assigned a node label indicating the trends direction ("+"or "-"). A frequent pattern mining based on the graph with these dynamically labeled nodes leads to common subgraphs which occur in limited time intervals (Trend Motifs). Figure 2.3 shows an example of identified motif occurrences, displaying for example opposing trends in global market share between the USA and Mexico, Argentina and South Africa (b).

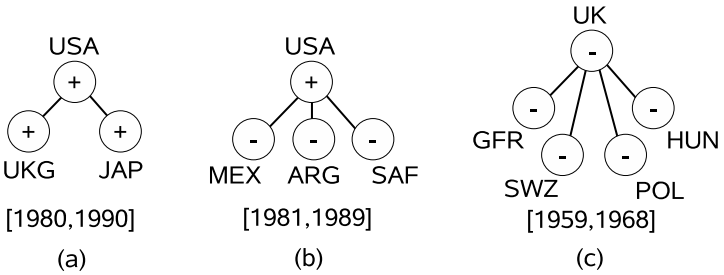


Figure 2.3: *Trend Motif* examples from [Jin et al., 2007]

We assume that extending this approach to allow for trend labels for edges instead of nodes could lead to more expressive patterns and a more fine-grained analysis which may be of use in anti-money laundering and fraud detection.

[O'Madadhain et al., 2005] focuses on predicting future event co-participation of entities (link prediction) and the change in importance of nodes over time. Link prediction is done by embedding the local graph structure and covariates in a fixed-dimensional feature space, allowing to use standard probabilistic classifiers on the binary label "co-participating", "not co-participating". The model is learned on historic data and applied to recent data. The event-based ranking is done by initializing each node with an equal amount of potential and propagating some potential from nodes not participating in an event to nodes participating in events at each time step. This leads to a higher potential and therefore higher importance for nodes which often participate in events over time.

A completely different link mining problem considering the time dimension is described in [Kubica et al., 2003]. Initial situation is a noisy graph, the goal is to find the "true" underlying structure. A random walk model is used for link generation, weighted counts of co-occurrences are then used to approximate the "real"link weights. A temporal weighting determines to which extent older links are counted a relevant. Weighting can also depend on the type of a link, allowing for link types which are more significant for the underlying structure than other link types. This approach could possibly lead to more coherent results than just omitting weak links when trying to find the constituting structure of a graph. The application in anti-money laundering analysis could therefore be of use.

Link Analysis is a huge and very active research area. Surveys as [Pottenger et al., 2006] give an overview of recent approaches and solutions.

2.3.5 Fraud Detection Based on Prior Model Knowledge

The existence of prior model knowledge reduces fraud detection to a retrieval problem of known patterns. Depending on the complexity of patterns of interest, approaches may make use of straightforward *SQL*-Queries, rules in a knowledge base, or graph pattern matching algorithms. Of course, those methods may be combined with other approaches which try to complement and refine the model knowledge. It is worth noting that in spite of a strong focus on (supervised) data mining approaches in purely academic experiments, applied research often relies on prior model knowledge, as can be seen in the following section. After introducing the field of relation based approaches in Link Analysis, we will stick with the focus on the relational prespective in this section, that is, on graph pattern matching algorithms. We first introduce a number of existing detection systems making use of graph pattern matching and subsequently look at the problem of graph pattern matching in slightly more detail.

Pattern Matching Fraud Detection Systems

The *NASD Regulation Detection System (ADS)* [Kirkland et al., 1998; Kirkland et al., 1999; Senator, 2000], the *Financial Crimes Enforcement Network AI System (FAIS)* [Goldberg and Senator, 1995; Senator et al., 1995; Kirkland et al., 1998] and the *Link Analysis Workbench (LAW)* [Wolverton et al., 2003; Berry et al., 2004] are systems which are reported to make use of pattern matching algorithms. *ADS* is built for monitoring trades and quotations in the *Nasdaq* stock market to identify patterns and practices of behavior of potential regulatory interest. It combines detection (pattern matching) and discovery (data mining) components and visualizes the results in various ways. Discovery is done by means of parallel association rule and decision tree implementations. However, details about the way those algorithms are used are not given. Pattern matching algorithms are available for both temporal sequences and defined rules. The sequence matcher works, like a regular expression matcher, with state machines. Users can specify patterns using a temporal sequence pattern editor. The rule matcher produces so-called “breaks”based on the “detection of repeated instances of predefined behavior”represented as rules. It is mentioned that the patterns are represented as conjunction tests on attributes and trees are created containing the pattern conjunctions with the number of times that a conjunction is detected. Tree traversal based on attributes tests is then used to match the pattern. Publications on the *ADS* are not focused on

the details of the pattern matching approaches, but on the whole system and its embedding in the regulatory processes. The system contains extensive management components administered by extensive human manpower. A knowledge management board presides over domain teams which discuss alerts and pattern configurations in weekly meetings with a *KDD*[17] specialist. A *KDD* team documents patterns and the related scenarios. Visualizations and pattern generation editors are not discussed at length in the publications, but it appears that their complexity requires elaborate training for users. *This points out that a detection system in this order of magnitude in productive use constitutionally focuses on the knowledge engineering and not on the sophistication of algorithms, which agrees with our observations*.

The *FAIS* System is a money laundering detection solution. Input data is a collection of large cash transaction reports made to the *U.S Treasury Departement* according to terms of the *Bank Secrecy Act* . One of the major tasks of the system is to link transaction reports to a distinct individual in the presence of errors, uncertainties and inconsistencies in the data. This consolidation step seems to rely initially on common identification information (for example, social security number or drivers license number). A more elaborate approach of consolidation is described in the related paper [Goldberg and Senator, 1995].
The system is used in two different modes. The user-driven mode allows for ad-hoc queries, while the data-driven mode is used to display subjects or accounts which show relatively high suspicion scores based on predefined rules. Link analysis in *FAIS* seems to be mainly a human-powered approach based on visualizations, which is a form of visual data mining.

The *LAW* system, which is also in productive use, is built for detecting terrorist and other criminal activity. The authors point out three possible approaches for this problem: the matching of known patterns of interest, the identification of anomalies where (legal) patterns are violated, and the discovery of new patterns of interest. *LAW* relies on the pattern matching approach, which is not a trivial problem, as the authors argue. Inexact pattern matching is provided as a pattern is defined as a prototype of a situation with allowable deviations. Patterns representations are based on graphs, and inexact matching is based on topological edit distance and ontological distance as an ontology of occurring concepts is available. Pattern specifications can be nested. The matching algorithm is based on *A*-Search* where a state in the search is a partial match and the cost is the

[17] KDD means "Knowledge Discovery in Databases" — put simply, a data mining specialist.

sum of the delete costs for yet unmatched nodes and links, and replacement cost of the current node and link mappings. For the expert evaluation of the found structures, this system also relies on visualization. The authors argue that this approach should not be pursued at the exclusion of others and therefore does not claim to solve all detection problems in this domain. Matching in the time dimension seems not to be regarded in the *LAW* system.

CrimeNet Explorer [Xu and Chen, 2005] is a system using social network analysis and visualization in combination with hierarchical clustering to assist identification of subgroups in criminal networks, however not accounting for the time dimension. The focus on subgroups in criminal networks suggests that this approach makes use of prior model knowledge while having an explorative character.

Graph Pattern Matching

In the following, we will focus on graph pattern matching approaches which are of special interest for the project at hand as the data can be seen as a multigraph $G(V, E)$ where V is the set of accounts (or customers, respectively) and E is the set of financial transactions. Graph based pattern matching has received considerable attention recently, in particular in the scope of next generation databases dealing with *XML*, Web, network directories, and structured documents, which often model the data as trees and graphs [Shasha et al., 2002]. It has applications in numerous areas as computer vision, biology, electronics, computer aided design, social networks, intelligence analysis and more. As extensive survey material of this very broad field exists [Shasha et al., 2002; Gallagher, 2006a; Gallagher, 2006b], we focus on a general overview and point out aspects which are of special interest in the context of our problem setting. A definition of the graph pattern matching problem and possible variations is given in [Gallagher, 2006a]. Graph pattern matching accordingly builds upon two elements:

1. A data graph $G = (N, E)$, where N is the set of vertices and E the set of edges in the graph and $e \in E$ is a pair (n_i, n_j) where $n_i, n_j \in N$. The vertices and/or edges of G may be typed and/or attributed.

2. A pattern graph or pattern query $P = (N_P, E_P)$ which specifies the structural and semantic requirements that a subgraph of G must satisfy in order to match the pattern P. A graph G'

is defined as a subgraph of G if and only if $N' \subset N$ and $E' \subset E$).

The task is to find the the set M of subgraphs of G that match the pattern P, where the notion of a "match"varies in different approaches. Variations are identified in the following dimensions:

1. Graph properties

   Basic graph properties are directed or undirected edges, weighted or unweighted edges , single or multiple edges between nodes and the existence of self-loops. Nodes and edges may be typed and have attributes. (A special case of typed graphs are semantic graphs. In semantic graphs, vertices represent concepts and edges represent relationships between concepts. Semantic graphs are based on an ontology which specifies the possible concepts, the relationships allowed between each pair of concepts and the attributes associated with each concept and relationship.)

2. Matching type (structural vs. semantic)

   A matching approach may only take structural properties into account (e.g., matches only on structural similarity). The inclusion of information about node and edge types and attributes leads to a often more application oriented approach.

3. Matching strictness (exact vs. inexact)

   A matching approach may only return matches which are in exact accordance with P. Inexact pattern matchers in contrast allow for differences between P and its matches to a defined extent for example to facilitate the matching in the presence of noise. For specific applications, matches which are similar to a defined query pattern may be of interest and exact pattern matches may be extremely improbable. This is also true for our setting. Some approaches allow for only partially defined and therefore imprecise query patterns (e.g. containing wildcards).

4. Optimality

   As for all search strategies, solutions may be optimal or approximate. While optimal searches are guaranteed to find a correct match (if it exists) with an exponential worst-case complexity, approximate approaches omit this warranty in trade for a lower complexity (often polynomial).

Gallagher [Gallagher, 2006a] additionally differentiates between single-graph (one large graph) and graph-transaction settings (denoting a set of relatively small graphs or, more precisely, graph components) as introduced in [Kuramochi and Karypis, 2005].

The basis of numerous approaches and extensions in graph pattern matching is the subgraph isomorphism algorithm introduced by Ullmann [Ullmann, 1976]. It enumerates all possible mappings of vertices in the pattern graph to vertices in the data graph using a depth-first tree-search algorithm. Each path from root to leaf in the search tree represents a complete mapping of P in G. Any such mapping that preserves adjacency in P and G (vertices that are neighbors in P map to vertices that are neighbors in G) represents an isomorphism from P to a subgraph of G.

The problem of subgraph isomorphism is known to be *NP-complete* [Washio and Motoda, 2003]. In large graphs, search space reduction is therefore crucial. A common strategy is known as *candidate selection*, where unpromising portions of the data are filtered out in a preprocessing step and the matching is done on the reduced graph. Candidate selection is typically based on calculated metadata which summarizes graph properties. Examples of such summary information are graph invariants. Graph invariants are e.g. computed based on vertex degrees [McKay, 1990] or on paths in the graph [Giugno and Shasha, 2002]. Two identical graphs will have the same invariants, the converse is not necessarily true. Comparisons between invariant values of data and pattern graphs can therefore be used to prune impossible solutions from the search space. Metadata can also be used to guide the searching process, e.g. based on selectivity estimates on partial pattern matches. In his survey, Gallagher [Gallagher, 2006a] states that effective candidate selection in semantic graphs is possible without generating graph invariants since they already contain rich data on which to filter potential matches.

GraphGrep is a semantic matching approach using path-based meta information for candidate selection, *TMODS* [Greenblatt et al., 2005; Coffman et al., 2004], and *TRAKS* [Aleman-Meza et al., 2005] are other semantic matching approaches, to name prominent examples. [Tong et al., 2007] is an example of a recent graph pattern matching approach for inexact best-effort matches. It allows indirect paths between nodes in G which are adjacent in P. The proposed method finds patterns in linear time on the size of the data graph. The algorithm makes use of candidate selection (called "seed-finder"), and expanding step which tries to find an adjacent, "good"matching node and a "bridge"function which tries to find a "good"path to connect two matching data nodes if they are

required to be connected according to the pattern graph. The algorithm assumes a single-edge setting an requires the graph to be represented as $n \times n$ node-to-node matrix.

To our knowledge , research on pattern matching approaches is highly biased towards unweighted single-edge graphs, where the existence of a relation between two nodes is a binary decision. We identified two requirements resulting from our business considerations which are typically ignored in the proposed approaches:

- **Matching in multigraphs with a time dimension**

  The *LAW* system mentioned above is one of the very few works which combine sequential and graph pattern matching. The search for patterns with defined properties on both the *network* and the *timeline* dimension (which can be seen as multigraph edges with a timestamp attribute) is an neglected perspective. Pattern matching approaches which incorporate node and edge attributes typically emerged from the conceptual graph research area, where time sequences play an inferior role.

- **Matching based on relations between edge and/or attribute values**

  The value of an edge may be of interest only in the context of other edge values, e.g. in graphs representing monetary transactions or phone calls. For example, a person A calling a person B at time t, which then calls 5 other persons later may be a pattern of interest for an anti-terrorism investigator only if all those calls happen very shortly after each other. The reviewed matching approaches, while allowing for inexact matches, require specific attribute values (which, of course, can be concepts). Pattern graphs where node or edge specifications depend on related attribute values in the graph[18] seems to have received very little attention, if any.

Furthermore, [Gallagher, 2006a] points out that most work is evaluated only on relatively small, synthetic graphs which are either completely random or regular. Performance on real world data is often not evaluated. Also, numerous approaches are built for very small individual graphs (where "individual graph"stands for graph components). Analyzing real world financial transaction graphs, we found that component size may range from very small to huge component sizes depending on transaction types of interest, customer segments and more. Additionally, existing

[18] For example, a pattern based on concrete attribute values may specify that one node in the pattern has the attribute value "5" and its direct successor has attribute value "7", while a pattern based on attribute value relations may specify that a node in the pattern has an attribute value that is "half the size of its successors attribute value".

candidate selection and indexing strategies are focused on graph structure and not on the combination of attributes, type and structure, which is assumed to have further optimization potential [Gallagher, 2006a].

2.3.6 Visual Data Mining

Visual Data Mining is another huge research area and is occasionally used in fraud detection. An example for the visual exploration of atypical behavior in financial time series data using two-dimensional colormaps is given in [Ziegler et al., 2007]. The colormaps encode growth rates of assets for all possible combinations of time of purchase and time of sale in a single figure. In comparison to traditional line charts, this approach simplifies the identification of "against-the-market"behavior, underperforming assets and similar. It is imaginable that this technique may also be valuable in fraud detection.

In the recent publication [Huang et al., 2009], a visualization approach is proposed for fraud detection in the stock market. The authors point out that most of visualization based systems do not consider the "analysis of related social networks"and that there has been little research into understanding how fraud detection systems can be designed to assist the cognitive process of the human analyst. A combination of 3D-treemaps for visualization of current stock prices, market capitalization and changing direction is proposed for a first stage, a second stage makes use of a network visualization in combination with pattern matching based on historical data. This is another example of a system using the frequent combination of model knowledge with visualization. It is not stated if the system has actually been implemented and tested on real data. However, practical issues are not considered in the publication at all, although stated prerequisites are extensive, for example the collection of patterns from past cases, which already turned out to be infeasible in our case.

WireVis [Chang et al., 2008] is another complex system providing numerous visualization options for the visual analysis of financial wire transactions. The authors report that expert evaluators were not able to use the system because of security-related hardware and software restrictions that made installation impossible so far. This again demonstrates the importance of accounting for practical limitations.

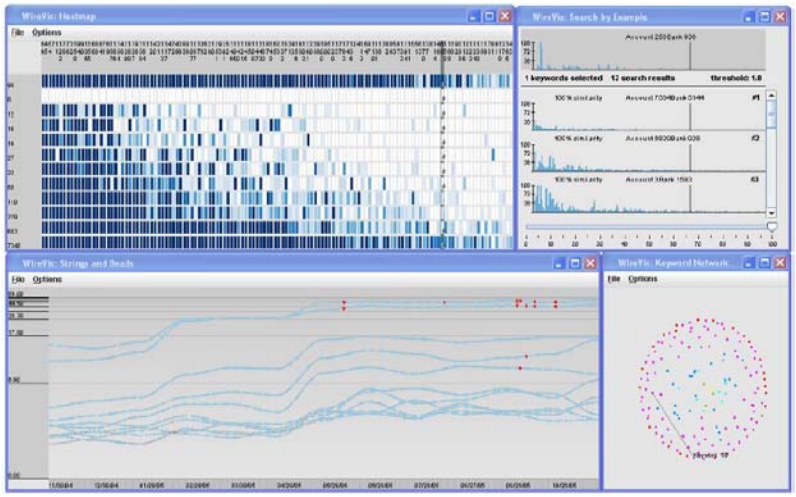


Figure 2.4: *WireVis* screenshot from [Chang et al., 2008]

2.4 Fraud Detection Approaches — A Unifying Framework

While evaluating the applicability of existing work for the situation at hand, we repeatedly found that considerable assumptions concerning the basic conditions were made, but typically not explicitly stated. In the following, we try to identify critical factors that have to be considered to find a feasible and appropriate solution to a specific fraud detection problem. We describe the identified factors with some considerations and combine them in a decision supporting framework. After this, we apply the framework to our situation.

We propose three factor types:

- **Company factors**

  form the situation given by the company, its properties and employees. Identified company factors are

  - *Prior Knowledge*
  - *Data Accessibility*
  - *IT Landscape*
  - *Computational Resources*
  - *Expertise Level*
  - *Evaluation/Refinement Resources*

- **Fraud factors**

  originate from the fraud type and its properties. We propose the following factors:

  - *Visibility level*
  - *Change level*
  - *Volatility level*

- **Population factors**

  are based on the monitored population and its properties:

  - *Inter object heterogeneity level*
  - *Intra object heterogeneity level*

Note that the proposed framework is not supposed to be exhaustive, but addresses the main issues we found to be of importance when designing our approach.

2.4.1 Basic Preconditions

A number of conditions are mandatory for the successful application of analytic fraud detection approaches. If they are given can be a matter of assumption. We identified the following preconditions:

- *A minimum of machine-readable data representing relevant information is available*. For illustration, log files of phone calls are a good starting point, while art forgery or marriage fraud may not leave any relevant machine readable data.
- *Fraud leaves a minimum amount of traces in the available data*. Traces are at least strong enough to distinguish a subset where the probability of fraud is reasonably higher than in the base population.
- *The introduction or extension of a fraud detection solution is economically advantageous*. Losses (e.g. monetary and reputational) which can be avoided must be in due proportion to the costs of the system, including its usage and maintenance. However, quantification of this ratio may be very hard or virtually impossible while being a required basis for decision-making.

2.4.2 Company Factors

Prior Knowledge

As mentioned above, the level of prior knowledge is a crucial design factor. In the case of a low level of prior knowledge, unsupervised, explorative approaches can be used to find potentially interesting (say e.g. uncommon) patterns. Those patterns can support the constitution of model knowledge when evaluated by human experts. Examples are [Kim et al., 2003b; Weston et al., 2008; Major and Riedinger, 2002; Brockett et al., 2002; Hollmén and Tresp, 1998]. If (formal) identification knowledge is available, supervised data mining approaches such as [Fanning and Cogger, 1998; Phua et al., 2004; Chan et al., 1999] can be used to calculate a model. In the case of informal identification knowledge, a conversion into a machine readable format has to be considered, but may not always be feasible depending on the nature of the information. Extensive informal identification knowledge can possibly be condensed and converted directly into model knowledge . As mentioned above, prior model knowledge motivates the retrieval of patterns which are known to be of interest [Kirkland et al., 1998; Wolverton et al., 2003].

Data Accessibility

While considerations on this factor are irrelevant for purely academic experiments, it is crucial if the solution should actually be of use in industry. An extreme case, which is nevertheless common for commercial vendors, is that the developers of a fraud detection system have no access to the data the system is intended to work with. This situation may complicate the development of a tailored solution in general and motivate approaches which can be seen as "configurable blackboxes". The design of the algorithms in the blackbox is not influenced by the specific situation at hand — only configuration is tailored. This may limit the consideration of individual conditions. However, we did not examine this issue in detail during the project at hand.

Of course, the amount and quality of available data is always an issue when doing data mining and related tasks. Special attention should be given to possible discrepancies between the data available to the detection solution and the information available to investigators. This discrepancy may be based on access regulations or on the lack of machine readable representation of the information. For example, in our case, extensive data describing accounts and transactions was

available, but free text information describing the customer relationship history was out of scope in spite of its importance for the evaluation of suspiciousness. The bigger this difference is, the less it can be the goal to find actual fraud cases, but to find patterns which are relevant for human experts. This may have some impact on design decisions.

IT Landscape

Constraints may exist that make a wide range of possible approaches infeasible or at least not directly applicable (for example [Chang et al., 2008]). A major topic is the reformatting of data, which is often required (e.g. [Holder et al., 1994]), but may be prohibited by company regulations, in particular for highly sensitive data. Some approaches are based on changes or extensions at the database system or even hardware level which may be impossible in an existing IT landscape. Only specified programming languages, or even only parts of a programming language may be allowed to implement the system.

Computational Resources

Computational resources can be a significant cost factor within the internal billing system of a company. Therefore, computational capacity may be limited. The complexity of analysis has to conform to the amount of data and the available resources.

Expertise Level

Generally speaking, approaches which are fully comprehensible for the users have a higher chance to be accepted, as intuitive work with the system is possible. For example, a fully-fledged decision tree may be meaningful for a technically interested user, but out of the question for others. As user acceptance is a critical success factor, the complexity of the system has to be in line with its users.

Evaluation/Refinement Resources

Several approaches envision refinement cycles where a human expert evaluates patterns returned by the system and feeds back findings into the system for refinement [Kim et al., 2003b], [Major

and Riedinger, 2002],[Senator, 2000]. This requires considerable effort from investigators which is a substantial cost factor, if disposable at all. Depending on the importance that is attached to the project on the part of industry partner management, disposable resources may be low. This may be the case in particular at early stages of a research project. In the case of low prior knowledge, extensive evaluation (and therefore learning) effort is indispensable, but tends to be more laborious than evaluation results based on model knowledge as the scope of returned patterns may be broader. Generally speaking, the lower the evaluation and refinement resources are, the more precisely should the notion of "fraud" be specified for a detection system to be successful. However, this, in turn, increases the risk of defining "fraud" too narrowly and producing false negatives.

2.4.3 Fraud factors

Visibility Level

Depending on its type and the used view on the data, fraud can be evident or invisible. We can differentiate between three levels of granularity — the *single event* level, the *aggregated event* (or *single object*) level and the *relation* level .

On a single event level, isolated instances are monitored (e.g. [Major and Riedinger, 2002]). An example is the monitoring for transactions which exceed a certain threshold. This assumes that a single event contains sufficient evidence. Limitations of this level are events which are part of a fraudulent behavior, but do not differ from normal transactions without the context of preceding and following transactions (e.g. when smurfing is used).

While a single object level is also common, it typically contains aggregated event level attributes [Fawcett et al., 1997] [Fawcett and Provost, 1996], e.g. in the case of financial transactions or phone calls, why we define it as a different granularity level. The aggregated event level omits the focus on single events by aggregating them over a certain time period. An example are daily or weekly account summaries. Given an adequate time window for the aggregation, suspicious activities like smurfing can be captured using aggregation. However, it increases the risk of masking interspersed fraudulent events which do not have sufficient impact on the aggregated values and vanish in the information loss of aggregation. We have met cases where employees interspersed inconspicuous transactions over a long period of time, remaining invisible in the high and varying transaction activity of the victims. The authors of [Whitrow et al., 2009] analyze the strategy of transaction aggregation in credit card fraud detection in detail.

In the relation level, events are analyzed in relation to other events and objects. We distinguish two dimensions:

- Network relations
- Time relations

The approaches in the link mining and graph pattern matching area work with this granularity level (e.g. [Holder et al., 1994; Berendt, 2005]). The high expressivity of the relation level may lead to the view that it is generally preferable to "lower" levels. However, it has to be considered

that results produced by relation based approaches tend to be demanding to evaluate and the approaches are limited by the fact that only part of the whole network may be available to the detection system. Consider a small financial institution which has only access to its own data while e.g. a money launderer has access to the whole financial system worldwide. In addition, the nature of the data may not be appropriate for a relational perspective as it cannot be represented as a graph.

Change Level

Reviewing the variety of proposed detection approaches in literature, we can clearly identify two different implicit assumptions:

- Changing behavior is a good discriminator for the detection of fraud
- Absolute indicators[19] are good discriminators for the detection of fraud

We can find an illustrative example to explain the difference in the area of mobile phone fraud. Consider a mobile phone which has been (legally) used for years and is cloned by a fraudster at time t. Lets assume the fraudster expects the account to be detected as defrauded and closed after a certain period of time (e.g. when the true owner is checking the monthly bill). In consequence, he/she will try to get as much as possible out of the account. In this situation, we can reasonably assume that the superimposed calls will lead to a significant change in behavior of the defrauded account (as for example done in [Weston et al., 2008]) and according approaches allow for detection at an early stage. Similar situations are imaginable in credit card fraud.
However, consider a fraudster which is not mainly interested in monetary profit from a cloned phone, but in masking his identity using a stolen phone number. A more subtle interspersion of fraudulent calls may not lead to a significant change in behavior. Another issue is phone subscription fraud where the fraudulent calls are present from the beginning (and possibly even exclusive). In those situations, absolute fraud indicators may be of higher value as for example applied in [Brockett et al., 2002].
Therefore, detection approaches based on behavior changes may be more adequate for fraud cases where the detection has relatively little consequences for the perpetrator (e.g. exclusion of using

[19] Indicators that do not track changes in behavior, but absolute attributes of an instance, e.g. "turnover at time t".

the defrauded resources without being identified), while absolute fraud indicators may be a better choice for detecting more cautiously conducted fraud.
Similar considerations can also be done in other fraud types and related topics as e.g. anti-money-laundering.
We challenge the general statement given in [Burge and Shawe-Taylor, 2001] that one of the most common indicators of fraud is a significant change in behavior. We rather propose to examine this assumption critically depending on the type of fraud and the possible motivation of the fraudster.

Volatility Level

Fraud volatility denotes how agile fraud reacts to changing circumstances (such as the introduction of an effective fraud detection system). A high volatility may correlate with high professionalism and high profit fraud. Developing new fraud schemes can be seen as a creative process which requires considerable effort and knowledge. For professional, organized crime (e.g. money laundering) it may be profitable to constantly adapt its processes to outmaneuver investigators, while for the crime of opportunity or crime out of a desperate situation the information exchange and sophistication level needed for a high adaptability is expected to be lower. High fraud volatility shortens the time a model is accurate and therefore requires a high adaptability rate of fraud detection solutions alike, which motivates the use of outlier detection methods, as labeled data from the past may not be adequate for modeling fraud in the future. This, in turn, implies considerable evaluation and refining resources from experts.

2.4.4 Population Factors

Inter Object Heterogeneity Level

The main differentiation we found in related work in terms of the examined population is based on heterogeneity. While some approaches rely on a global model for fraud based on the whole population, others argue that a significant (behaviour) heterogeneity between different objects (e.g. accounts) motivates the calculation for similar groups or even individuals. Local models increase the complexity of the system, but are valuable in situations where events may be suspicious for one instance while being completely reasonable for another instances. [Cahill et al.,

2002] states that "classifying accounts into segments and applying thresholds to each segment separately may improve performance, but at the expense of multiplying the number of thresholds that have to be managed".

Intra Object Heterogeneity Level

In [Fawcett et al., 1997], it is stated that "superimposition fraud is detectable if the legitimate users have fairly regular behavior that is generally distinguishable from the fraudulent behavior". Unfortunately, peoples situations and the way they make use of their resources are exposed to change. A credit card owner may go on holiday and exhibit completely different purchasing behavior in a far country, which is otherwise a possible evidence for a stolen credit card. The usage patterns of bank accounts may alter as their owners change living arrangements, switch wage accounts or have other, irreproducible reasons. As an example, unsupervised approaches will suffer from both a high level of inter and intra object heterogeneity. While approaches detecting behavior change will produce a substantial amount of false alerts in the presence of a high intra object heterogeneity, a high inter object heterogeneity may make it impossible to determine common behavior or to detect outliers for discrimination between fraud and legitimate patterns.

2.4.5 Conclusion

Based on the factors introduced above, we propose the following questions to be asked when considering a fraud detection problem:

- What is the form and level of prior knowledge available? (PK)
- What is the level of data accessibility?
- What are the constraints given by the IT landscape?
- What are the available computational resources?
- What is the (technical) expertise level of the users?
- What evaluation and refinement resources are (and will be) dispensable? (ER)
- What is the expected visibility level of fraud? (VL)

- What is the expected change level of fraud? (*CL*)
- What is the expected volatility level of fraud?
- What is the heterogeneity level of the population? (*HL*)

In table 2.3, we try to give a condensed view of what others have done in the according situation. Note that we ignored some of the questions posed above. While being relevant for the design process, possible answers to those questions can have a huge variety and are inappropriate for a crisp categorization. Considerations about those questions can be found above. Furthermore, we combined different parameter values where reasonable: The right choice of either a single event or aggregated event view is an important factor, but we found that approaches using one view should generally be readily adaptable to the other. While high evaluation resources are required in the presence of low prior knowledge, it is reasonable to state that for higher levels of prior knowledge, extensive evaluation resources are still very valuable, but not a strict requirement. Where approaches explicitly dedicated to either global or local solutions do not exist, we combined these values and added some hints in the remarks. We argue that a focus on behavior change implies a (temporal) relation visibility level. The opposite is not necessarily true. Temporal patterns can be of interest while the focus does not lie on behavior change but static patterns.

Prior Knowledge	Evaluation Ressources	Visibility	Change	Heterogeneity	Remarks
low	low				Problematic as building up knowledge is essential if not previously available
low	high	relation	change	global	For purely temporal relations e.g. a globalized version of Peer Group Analysis [Weston et al., 2008], for combination with network relations [Jin et al., 2007; O'Madadhain et al., 2005].
low	high	single/aggregation	static	global	Outlier detection, e.g. a globalized version of [Major and Riedinger, 2002], at a higher level of prior knowledge RIDITS [Brockett et al., 2002]
low	high	relation	static	global	For temporal relations, e.g. [Kim et al., 2003b], for network relations unsupervised graph-mining as e.g. [Holder et al., 1994; Jonyer et al., 2000; Eberle and Holder, 2007; Zaki et al., 2004], at a higher level of prior knowledge [Tong and Faloutsos, 2006]
low	high	relation	change	local	For (only) temporal relations, Peer Group Analysis [Weston et al., 2008; Burge and Shawe-Taylor, 2001]
low	high	single/aggregation	static	local	Outlier detection [Major and Riedinger, 2002]
low	high	relation	static	local	Localized versions of [Kim et al., 2003b; Stolfo et al., 1998] or *localized* unsupervised graph mining, which involves the problem of adequate graph partition
labels		single/aggregation	static	global	Traditional supervised DM, e.g. [Fanning and Cogger, 1998; Prodromidis and Stolfo, 1999; Maes et al., 1993]...
labels		relation	static	global	Graph classification as e.g. mentioned in [Holder et al., 1994], guilt-by-association approaches[Cortes et al., 2001; Macskassy and Provost, 2005]
labels		relation	change	local/global	For (only) temporal relations: Adaptive Fraud Detection [Fawcett et al., 1997; Cahill et al., 2002], Activity monitoring [Fawcett and Provost, 1999]. Global rules are used with local parameter values.
model		single/aggregation	stat./chng.	local/global	Rule-based filtering, e.g. SQL-Filters, for detecting change in combination with history from previous filter runs. Local specifications can be part of the
model		relation	stat./chng.	local/global	For network relations: graph pattern matching ,[Berry et al., 2004; Wolverton et al., 2003; Senator et al., 1995; Kirkland et al., 1998]..., for temporal relations sequence pattern matchers or adequate SQL-Filters. For both network and time relations graph pattern matching algorithms with temporal aspects [Kirkland et al., 1998; Senator, 2000; Kirkland et al., 1999]

Table 2.3: Fraud Detection Framework

2.4.6 Applying The Framework To Our Project

We now make use of the unifying framework to assess the situation we met and justify our design decisions.

- **Prior Knowledge Level**: *Model Knowledge available, Identification Knowledge not available*
  At the beginning of the collaboration with Alphafin, we were given a number of internal fraud case reports in an unstructured free text format the according machine readable data (fraudulent transactions, involved accounts) was neither specified in detail in the reports nor accessible for analysis. The limited amount of informal identification knowledge could therefore not be converted to machine-readable identification knowledge. However, model knowledge could be established in analyzing the reports and learning from fraud experts in extensive discussions. Furthermore, in the course of our project, Alphafin started to externalize fraud expert knowledge and use it for developing a *SQL*-based alert system. This resulted in further, well-specified model knowledge.
  Given this situation, we had to omit supervised data mining approaches. The availability of pattern matching approaches motivates the use of pattern matching.

- **Data Accessibility**: *Largely accessible, however at a late stage*
  In the early stages of the project, we had to rely exclusively on the model knowledge given by fraud experts. Not until designing and implementing part of the proposed system — specifically the visualization component — were we given access to a large amount of relevant data, which was crucial for numerous detail decisions and configuration of our approach. The fact that data access for model calculation was initially not given and its future accessibility remained unclear for a long time further motivated the focus on available model knowledge.

- **IT Landscape Constraints**: *Java, SQL, no reformatting/ local copies*
  We were constrained to Java and data access via basic SQL (no PL/SQL, stored procedures or similar). Reformatting the data was out of scope not only because of the prohibition of local copies but also by the requirement to design an approach which is directly applicable to the whole data warehouse. Furthermore, any changes on the database or hardware level were prohibited. This disqualifies the use of existing systems making use of special formats

as e.g. *Subdue*.

- **Available Computational Resources**

  As the database system is concurrently used by various applications and users, database answer time was subject to a considerable variance due to changing workload. Furthermore, we had no influence on index structures and data organization, which was optimized for other productive applications. In the presence of the massive amount of data and the limited computational resources on the client, the chosen approach had to be of adequate computational complexity to reliably produce results within a reasonable amount of time.

- **User Expertise**

  From beginning, we tried to build our approach as close as possible to the existing business processes and concepts with which potential users are already familiar. This allows for a straightforward integration into existing solutions and can decisively increase acceptance of a novel approach, which is a critical success factor. This design decision disables the use of solutions which are hard to comprehend or to evaluate. We found that even in the case of a straightforward network component, suspiciousness assessment can be a task of considerable complexity. Our goal was to find an adequate tradeoff between expressivity and understandability.

- **Evaluation and Refinement Resources**: Low

  The high workload of fraud experts and the potential complexity of the evaluation process suggested to be prepared for low evaluation and refinement resources during the research stage. As described in later sections, our intentions to have fraud experts generate identification knowledge while using our system was not successful due to the lack of manpower resources. This experience affirmed our decision to omit approaches which absolutely require extensive evaluation and refinement — in particular, unsupervised mining approaches and very broadly specified pattern retrievals.

- **Fraud Visibility**: Relational

  No doubt a number of evidences of internal fraud is best captured on the single event or aggregated view. As mentioned above, we argue that detection of more subtle patterns of internal fraud requires a relational view on both the network and temporal dimension. At

this point it is worth to note that in our case, fraudulent activity is less single object-centered as other forms of fraud may be. Consider an external fraudster which gains access to a user resource. Fraudulent activities will originate from the defrauded object/ node. The internal fraudster in contrast has access to a whole set of objects/nodes in the network by default (i.e. the advised customers) — a fact that may lead to completely different, more cross-linked patterns and to a setting where relevant patterns emerge within company borders with high probability. This motivates taking a relational view of the problem. We therefore focus on a relational perspective to optimally complement *Alphafins* own detection system, which covers the single event and aggregated event level.

- **Fraud Change Level**: Static

  As an exposed employee fraudster will suffer serious consequences, internal fraud is expected to exhibit more subtle patterns as a stolen credit card, which is a prominent example for behavior-change based approaches. The considerable intra object heterogeneity (see below) may further complicate the use of a change-sensitive detector. However we do not argue that change based systems are inappropriate. While focusing on a static approach because of the reasons mentioned above, we propose the extension of our approach from a static to a change based view and develop a possible design in future work.

- **Population Heterogeneity**

  We did not conduct a systematic and exhaustive quantitative analysis of the population heterogeneity, but both discussions with human experts and our findings when building up business and data knowledge suggest that both intra object and inter object heterogeneity is high. The wide range from private retail customers to corporate or institutional customers directly indicates a high inter object heterogeneity. This motivates the use of local models. However, the relational view motivated by the fact that the internal fraudster is a meta-user of the network (see above) may have some non-trivial influence on these considerations. In terms of intra object heterogeneity we found that strong and sometimes abrupt changes, for example a complete inactiveness followed by irregular and short periods of excessive activity are repeatedly observable. We discussed such effects with human experts, but were not able to examine them extensively. The flexibility in terms of switching between local and global models and adapting scorings accordingly further motivates the use of pattern

matching techniques.

3

Solution Approaches

As argued in section 2.4.6, the situation at hand motivates the focus on relational pattern matching. To set the stage for a detection component, the development of a evaluation component was required. In this section, we will first justify and introduce our implementation of the evaluation component, which is a visualization tool. After that, we introduce the different approaches developed for the detection component. The interaction of the two components is discussed in detail in the case studies in chapter 4.

3.1 The Visualization Component

3.1.1 Justification

In section 2.1.2 we established the focus on relational patterns (in particular Transaction Chains and Smurfing) in our approach. We have also mentioned that human experts require adequate visualization to efficiently and effectively find and interpret such structures. Visualization is therefore crucial for the envisaged approach.

One might ask why the undertaking of designing, developing and launching another visualization tool was a critical factor for the project. After all, as mentioned before, fraud experts already made use of commercial visualization software.

Using the existing visualization tool however was prohibitive due to the following reasons:

1. The product was only licensed for very few specialists. The approval for another, very costly license for research was highly improbable in the given situation.

2. The initial business regulations we were confronted with did not allow us to access the data directly without a productive tool, and the manual process of ordering data from employees having access was too resource- and time-consuming for research.

3. It soon became clear that the commercial system lacked crucial features (such as a direct connection to the data warehouse) and we needed a flexible and extensible solution which allowed adding functionality to integrate research findings. This would have been impossible when relying on a closed source commercial tool.

The consequence was to build a flexible visualization system (called *TVIS*[1]) which allows fraud experts (and researchers likewise) to directly access the relevant data and use the salient human pattern recognition abilities. TVIS became the linchpin of our analytic detection approach. It is not only used to evaluate the results produced by the detection component, but also as a convenient aid to mediate the assessment of data on the basis of relational and sequential patterns to extended stakeholders and to demonstrate the functionality of the detection component to business experts. "This pattern is new to me, what does it represent? How did it evolve?" These or similar questions arise when working with TVIS in both ad-hoc and result evaluation mode. Trying to find an answer to those questions allows for building up new pattern-related knowledge and to develop a paradigm of "thinking-in-patterns".

3.1.2 Main Requirements

On a very general level, the identified main requirements for TVIS were

- Flexible data request based on a number of relevant attribute values for both nodes (customers) and edges (transactions) in the graph. Relevant attributes were previously identified in collaboration with business experts.

- Support for both ad-hoc and detection component result evaluation usage.

- Customizable graphical and tabular data views which preserve consistency on data changing operations

[1] TVIS stands for "**T**ransaction **VIS**ualization"

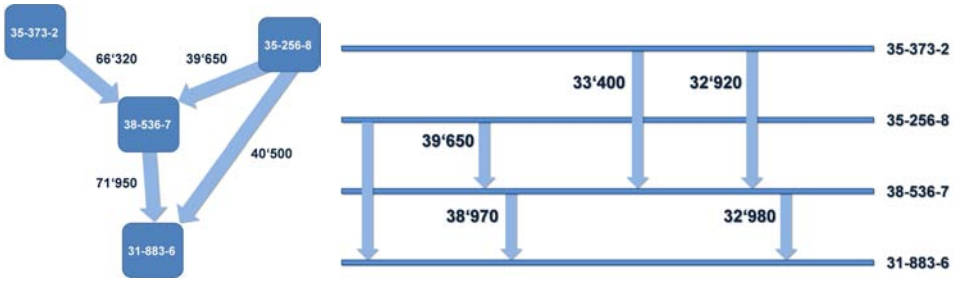
- The possibility to annotate relational patterns for pattern generalization/ supervised learning (labeling)
- A number of convenience features (Saving, printing, export functions)
- Intuitive handling and acceptance by gearing towards concepts and tools already known to future users

3.1.3 System Description

Visualization is a well known and widely used instrument in both research and industry. While the design and development of the system at hand was indispensable to set the stage for the research on detection approaches, it is of limited academic interest. In our opinion, an evaluation of work efficiency with TVIS would most probably boil down to the finding that structural patterns and transaction chains are more readily recognized visually than in the traditional table form for humans. This insight is neither new nor surprising - it can easily be reproduced by the reader by looking at Figure 3.1. We therefore only provide a high level description of the functionality TVIS offers and decided to forgo a systematic evaluation. More implementation oriented information can be found in [Appendix A]. The core of TVIS is formed by three different views on the requested data.

3.1.4 The Network View

The network view display the requested data in terms of a directed single-edge graph (see Figure 3.2). Nodes represent customers. As a customer can have more than one account, nodes contain nested graphs. In nested graphs, nodes represent accounts and edges represent carryover transaction between the accounts of the according customer.
Edge weights are defined by the sum of transaction amounts between the source and the target node within the defined time window. While transactions between *Alphafins* customers (internal transactions) are straightforward to display, transactions with an external counter party (external transactions, e.g. cash withdrawals or transfers to other financial institutions) are more challenging: the desirable distinct identification of the external party is bounded by the limited data qual-



(a) (b)

ID	ORIGINATOR	BENEFICIARY	AMOUNT	DATE
343732	35-373-2	38-536-7	33'400	05.10.2008
433624	35-373-2	38-536-7	32'920	10.10.2008
434624	35-256-8	31-883-6	40'500	17.09.2008
346377	35-256-8	38-536-7	39'650	25.09.2008
234637	38-536-7	31-883-6	38'970	26.09.2008
275745	38-536-7	31-883-6	32'980	11.10.2008

(c)

Figure 3.1: A small example of six transactions in a network view (a), a timeline view (b) and the traditional table view (c)

ity. It is often not possible to recognize recurring external transaction partners due to ambiguous free text identification. Two options remain: The generation of a new external node for each external transaction, or the definition of one or more external meta-nodes which assimilate all external parties. The two options were combined by using meta-nodes, showing an aggregated view of the amount of money entering or leaving the company, and allowing for a view on the single transactions and their external counterparties by "opening" the meta node. After experimentation with a single meta-node for all external parties, we decided on two role-dependent meta-nodes; one node representing external beneficiaries and the other representing external originators. This simplifies the display of funds inflow and outflow.

The network view allows for an overview of transaction connections, in particular for internal transactions, which are of special interest for internal fraud detection. In the quite common presence of recurring transaction partners, thousands of transactions may result in a clear and straightaway meaningful image. It is this single-edge graph view on which most of the present graph-pattern matching solutions are based on. However, as we argued before, sequential infor-

mation may also be crucial to make sense of transactional behavior. This motivates the second graphical representation, the timeline view.

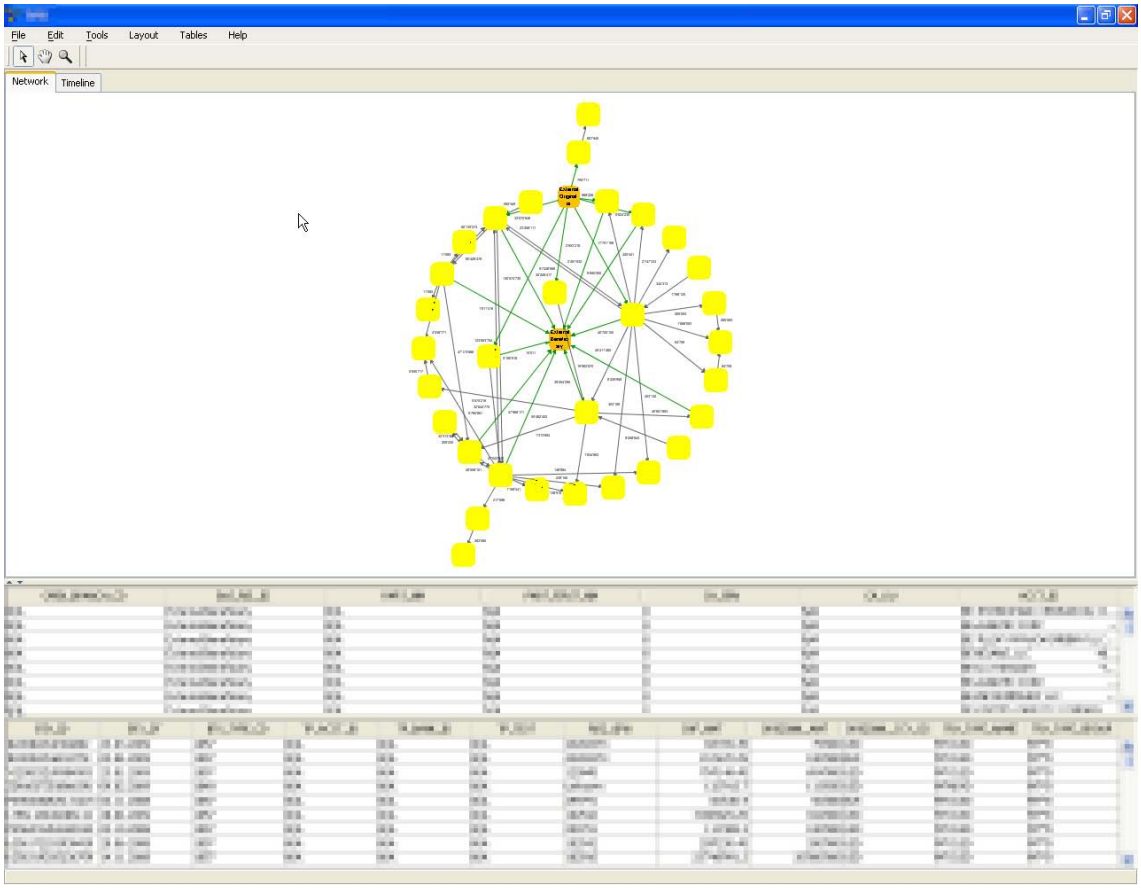


Figure 3.2: TVIS — the *Network View* (real data, confidential information is made anonymous and pixelated)

3.1.5 The Timeline View

The timeline view is focused on the sequence of single transactions. This view represents a directed multi-edge network where nodes are translated into lines along the time axis. Single transactions are placed on the time continuum between the lines representing the according customers (figure 3.3). Edge weights represent single transaction amounts (in a general base currency).The time axis can be stretched or packed. This allows for both a detailed analysis of single transactions in an activity intense setting and for a long term overview, where single transactions may merge and overall activity patterns are unveiled (for an example, see figure 4.5). Timeslots with no activity can be hidden, which is valuable in the presence of sporadic activity. Again, the external originator and external beneficiary meta nodes are used. The maximal amount of data presentable in this view is usually considerably lower than in the network view. In particular, the maximal number of customers is limited as the image loses clearness as it grows in its vertical dimension. Therefore, this view is intended to be used after the data has been narrowed down to structures of special interest using the network view.

3.1.6 The Tabular View

While nodes in the network are labeled with the according customer number and transactions are labeled with their amounts, more node and edge attributes are needed for the assessment of patterns. This additional data is provided in tabular view below the graphical view. Two individual panels are displayed for node attributes and edge attributes to keep the distinction between the two concepts intuitive. For example, additional node attributes may be customer name or customer type information, the assigned customer advisor and similar. Additional edge attributes may be the original currency of the transaction, transaction type and date or the triggering customer advisor. The displayed selection of node and edge attributes is configurable.

3.1.7 View Editing and Consistency

For extensive pattern evaluation, the views have to be configurable and changeable. Some graph elements may be safely discarded after a first glance. The user may become interested in additional graph elements which are not currently loaded and displayed. Moving through the total

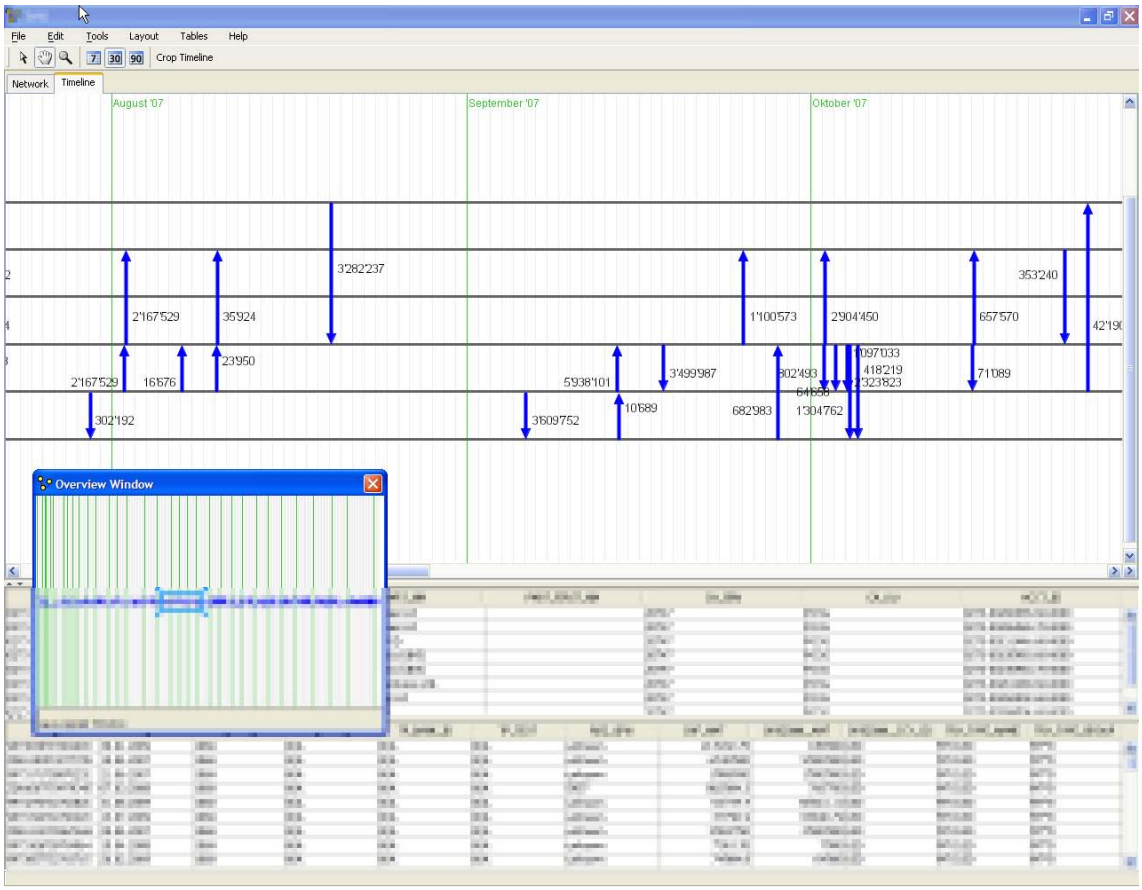


Figure 3.3: TVIS: The *Timeline View* (real data, anonymized and pixelated)

graph in terms of discarding parts and dynamically loading other parts into the ”cone of light” is crucial for an effective and efficient evaluation (figure 3.4).

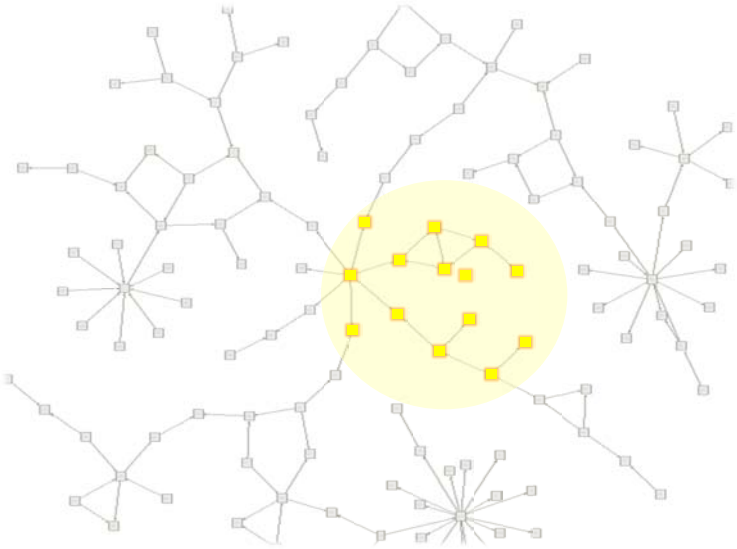
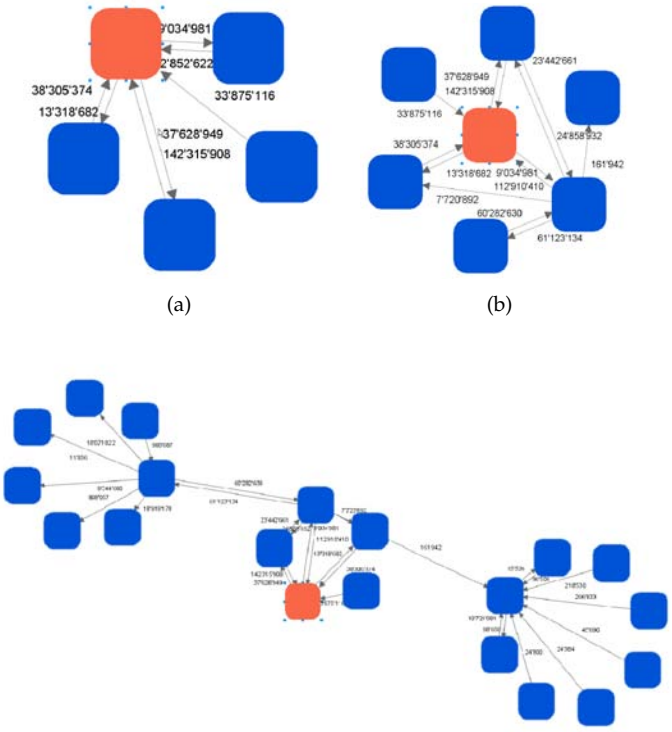


Figure 3.4: Like a ”cone of light”, TVIS reveals parts of the total graph. Navigation in TVIS corresponds to moving, enlarging or narrowing the light cone

TVIS allows for permanent deletion or temporary hiding of graph elements. The filter panel for requesting the data gives the option to add the newly requested data to the current visualization (instead of discarding it). A special case of this option is the neighbor search. It allows to load all direct transaction neighbors of selected customers and therefore facilitates the expansion of the network in directions of interest. For illustration, consider a customer with a — in some way — noticeable business connection with a company. The investigator may want to have a closer look at it; a neighbor search will add the transaction network of the according company and allow for a more detailed assessment. Figure 3.5 illustrates a real world data visualization primarily based on one customer (a), followed by two consecutive neighbour searches, incrementally revealing the direct and indirect neighbourhood of the customer (b) and (c). Selections and modifications can be made in any of the available views. Consistency between views is guaranteed by propagating those actions. As selection is propagated, mapping between graph elements and the according attributes in tabular view is straightforward.



(a) (b)

(c)

Figure 3.5: An example of a neighbour search. The primary customer of interest is marked red (real data, confidential information masked). Two direct neighbours exhibit further connections after the search.

3.1.8 Pattern Annotation

Whenever an investigator encounters a pattern of special interest, she/he can select it, annotate it with additional information and add it to a pattern repository, which is part of TVIS. The goals of the pattern repository are the following:

- Knowledge exchange between fraud investigators
- Reducing knowledge drain due to personal fluctuation
- Acquiring pattern based model knowledge for (automated) specification of generalized patterns
- Acquiring relational labels (that is, identification knowledge) for supervised data mining

After the release of TVIS it soon became clear that the buildup of a pattern repository of a reasonable size was an unrealistic goal under the given circumstances. This is due to two main reasons:

- The data warehouse only contained the data of the two most recent years at a time, older data was therefore not available for TVIS. Given the sparsity of revealed internal fraud cases, most of the eminent cases with numerous patterns were actually time-barred in terms of the system.
- Limited manpower resources made the explorative search for interesting patterns impossible. We therefore had to rely solely on expert interviews to externalize relevant model knowledge.

3.2 The Detection Component

Given the previous findings and the decision to focus on specialized pattern matching, a number of possible approaches and variants for the detection component were developed. In the following, each approach is discussed in detail.

3.2.1 The ChainFinder

The *ChainFinder* is an extension of an existing, well-known graph pattern matching technique, namely tree-based search (see [Chakrabarti and Faloutsos, 2006]). The algorithm is tailored to meet the following requirements:

- Matching of tempo-relational patterns in a multigraph.
- Matching of graph patterns with relational constraints on node or edge attributes
- Feasibility in the presence of strict technical constraints and massive amounts of data
- Straightforward, comprehensible functionality to ease acceptance and practicability for business users

More specifically, the main intent was to find a feasible way of retrieving Transaction Chains and Smurfing patterns as described and motivated above (section 2.1.2). The problem was solved by taking the well-known technique of search-based pattern matching and slightly extending it with the concept of information particles travelling along search paths.
In contrast to classic tree traversal, the goal state is not defined by reaching a certain node, but by the content of the particles, which accumulate information on traversed nodes and edges. Typically, not the individual node and edge attribute values are of interest, but their relationship to each other. The compatibility of attribute values are used for pruning the search space.
Therewith, the ChainFinder forgoes general sub-graph homo/isomorphism matching capability in favor of a intuitive chain and Smurfing pattern matching. It is worth noting that the ChainFinder, being a tree-search based approach, starts searching for the patterns in question at a dedicated root node at any one time. This procedure is reasonable if a subset of nodes in the graph is of special interest in respect to their role as potential sources (or intermediaries/target accounts

as mentioned below). For an unspecific overall search in the total graph for chains starting at any point, runtime as well as results may be exposed to considerable redundancy. Other approaches (as the *GraphSlider* mentioned below) may be favorable in this setting. However, in most application settings we met, the assumption of a subset of special interest was appropriate.
The mode of operation is illustrated in figure 3.6. The root node (a) is expanded by retrieving its successors (that is, in the default case, its beneficiary counterparties) (b). Particles are sent from the root node to the successors along the edges (i.e., transactions) in the search tree. The particles are encoded with transaction information of interest for the matching, for example transaction amount, triggering client advisor and similar (c). In the following, each of the successor nodes is expanded (d). The particles at the successor nodes are sent along a new edge if they are *compatible* with the transaction(s) in question (e,f).
Depending on the configuration, a particle may for example be compatible to a new transaction if

- the triggering client advisor matches, and
- the current transaction amount encoded in the particle is similar to the new amount, and
- the date of the last transaction in the particle is shortly before the new transaction date

When a particle is sent along a new edge, it is updated with information about the visited path while travelling. Hence, particles essentially encode information about chains of edges.
Recursively, any node containing particles is expanded and all its particles are investigated for further dissemination (g,h). Whether a particle is allowed to travel further always depends on the compatibility of information it contains with edges along which it is scheduled to proceed. If no compatible path is available, the particle ends its journey. It gets logged if the path it travelled is of sufficient, user-defined length (and is discarded otherwise) (i) . The algorithm ends when all particles have completed their journey. A number of issues and variants of the basic algorithm are considered in the following sections.

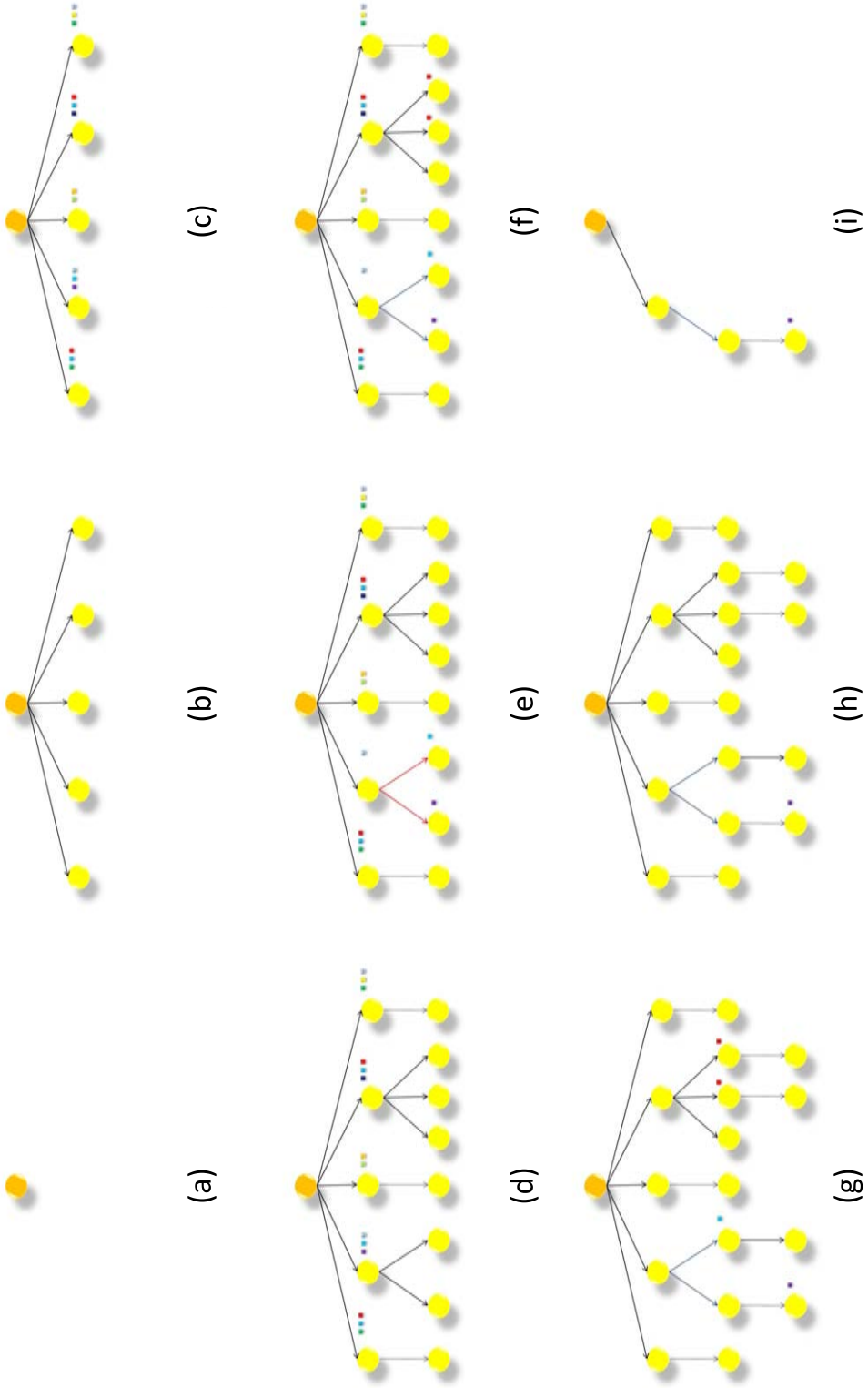


Figure 3.6: The ChainFinder algorithm illustrated

Algorithm 1 The ChainFinder Pseudocode

```
1:  /* set up data structures */
2:  for each successor node s of RootNode do
3:      s.particles ← buildParticles (n,s)
4:      if s.particles ≠ ∅ then
5:          push s on stack
6:      end if
7:  end for
8:
9:  /* search graph */
10: while stack non-empty do
11:     n ← pop stack
12:
13:     /* expand where particle is compatible /*
14:     usedParticles ← ∅
15:     for each s ∈ successor(n) do
16:         for each particle p ∈ n.particles do
17:             if compatible(p, edge(n,s)) then
18:                 p.path ← p.path ∪ edge(n,s)
19:                 s.particles ← s.particles ∪ p
20:                 if s ∉ stack then
21:                     push s on stack
22:                 end if
23:                 usedParticles ← usedParticles ∪ p
24:             end if
25:         end for
26:     end for
27:
28:     /* clear up remaining, non-travelling particles */
29:     n.particles ← n.particles \ usedPartciles
30:     for p ∈ n.particles do
31:         if p.pathlength ≥ minimumlength then
32:             log(p)
33:         end if
34:         n.particles ← n.particles \ p
35:     end for
36: end while
```

Where: `compatible(x,y)` encodes the traveling condition, `edge (x,y)` retrieves the relevant edges, `log(x)` writes a particle/path to the log, and `buildParticles(x,y)` builds the necessary particles from the first edge. text

Particle Cloning

A travelling particle can meet more than one compatible transaction for traversal at a node. As it cannot be decided which of the possible transactions is the "true" one for the chain, the particle gets cloned and an instance is sent along each compatible transaction. Therefore, a single particle can end up in numerous logged chains depending on how many times it was cloned during its journey. Intense cloning decreases the probability of "true single chains". This consideration is the basis of the *Random Structure Identification* introduced in the following section.

Random Structure Identification

First experiments with the ChainFinder revealed that repeatedly, false positives were retrieved where chain-like structures emerged obviously randomly out of intense transaction behavior. It was observed that these random structures commonly lead to numerous particle splits.

Consider a customer A transferring 5000 dollars to a customer B, which has hundreds of incoming and outgoing transactions each day (e.g. a big investment company). The probability that some of those numerous outgoing transactions will match the incoming amount of the transaction from customer A is very large, although the two payments are fully independent of each other. To decrease retrieval of such structures, the number of allowed splits (i.e. clones, see previous section) can be limited. If it is exceeded, the originating particle is discarded and logging is avoided. Of course, it cannot be guaranteed that this approach is not eliminating true positives embedded in an activity-intense context. While a maximum number of 3 splits showed reasonable results in our experiments, the optimal value is highly dependent on the underlying data and the configuration of the *ChainFinder*.

Single Chain vs Sum Chain Detection

Patterns previously referred to as Transaction Chains are retrieved in *Single Chain mode*, Smurfing patterns in *Sum Chain mode*. The main difference lies in the particle generation. In single chain mode, each individual transaction originating at the root node generates a discrete particle. In sum chain mode, particles are not generated for every single transaction, but for aggregated transaction sums within a defined time window. In the simplest case, all available transactions originating at a root node are summed up and one single particle is generated, holding the aggregated sum[2]. For smaller window sizes (for example a week or month), a particle gets generated for each propagation, summing up all transaction within the current window. From a visualization perspective, the single chain mode therefore refers to the timeline view where the sum chain mode rather refers to the network view. The choice of an adequate time window for the sum chain mode deserves a closer look.

Choosing a very small window (and propagation step) size may quickly lead to a prohibitively

[2]That is, if A paid, in total, 100'000 dollars to B over the whole period of time the data is available (in our case two years), one single particle is generated with code "100'000". If B paid a sum of approximately 100'000 dollars to C during this period, a chain of length 2 is registered.

large number of generated particles, increasing the probability of random hits to an unacceptable level. At a large window size, the algorithm may not be able to catch local smurfing patterns because of interfering transactions before or after the smurfing. Therefore, the right choice for the window size heavily depends on two factors :

- The typical timeframe for a smurfing pattern
- The degree of interference of legal activities with illegal smurfing (or the probability that a connection between two node contains both illegal and legal transactions).

Figure 3.7 illustrates a window size which is too small for the occurring smurfing. A simple solution to the problem of choosing the right window size is, as mentioned above, to aggregate over the total available interval, which is plausible under the assumption that smurfing is exclusive and not interspersed into other behaviour (or other behaviour is marginal enough to not bias the detection). We limited our experiments to this setting.

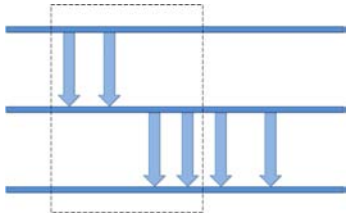


Figure 3.7: An example of an inadequate window size for smurfing detection

Timeline Scoring

Related to sum chain detection is the *Timeline Scoring* model that was implemented for further assessment of potential Smurfing. The underlying assumption is that in Smurfing, money tends to be transferred to an intermediary account before it is withdrawn, independently from the account balance. The timeline score assesses the accordance of the pattern with this rule. A simple implementation takes the aggregated amount of money which was withdrawn before being deposited as a negative score. The lower the score is, the lower is the probability that the according sum chain is smurfing in respect to the assumption above.

Search Direction

In the default case, the subset of nodes of interest (that is, the root set) consists of potential chain *sources*, i.e., particles start at outgoing transactions. The ChainFinder can be configured to start at potential chain *sinks*, so particles will start at incoming transactions of the root and travel upwards. A third possibility is the definition of *intermediary* nodes where particles are sent down- and upwards simultaneously. This slightly increases the complexity of the algorithm as the particles tracking the same chain in opposite directions have to communicate with each other to determine the chains current length. While theoretically equivalent, a particle travelling upwards turned out to be substantially slower than a particle travelling downwards in our setting, which is caused by the index structure of the database.

Reporting Date Loop Detection

When detecting chains, paths containing loops are eliminated. However, settings exist where the detection of loops is of interest. In section 2.1.2 , it was mentioned that a fraudster may temporarily try to balance missing amounts, in particular in the wake of an approaching known reporting date, where accounts are checked internally or by the customer. This can lead to the deposit shortly before and withdrawal shortly after a certain date. Accordingly configured, the ChainFinder can detect those patterns.
Smurfing may make use of several source/ intermediary and target nodes. At this point, patterns are getting very complex. The more complex and distributed a pattern is, the higher is the probability of false positives expected to be. While this may not be true for other fields, the assumption seems reasonable that in internal fraud, sophisticated, “faultless” smurfing patterns over distributed sources, intermediaries and/or targets are improbable. For simplicity, we decided to limit our experiments to single role nodes. All the same we developed a number of ideas and considerations as possible starting points for future work, which are discussed in the according section.

Particle Compatibility Configuration

The conditions on which a particle is allowed to travel along a path are crucial for pattern specification. This involves configuration of the maximum time span between two consecutive transactions, the maximal amount deviation allowed, and the minimal length of a chain to be logged after it stopped growing. In respect to the nature of chains in the present internal fraud setting we implemented an additional constraint checking if all transactions of a Chain or Smurfing are triggered by the same employee. Further compatibility constraints may easily be added, incorporating any checks on account or transaction attributes. For instance, it may make sense to set the constraint that the last transaction in a chain is ending outside the financial institue (at an external party) and/or is a cash transaction. The definition of further constraints may be part of a configuration refinement.

A word on implementation issues

Two different implementations were used for experimentation with real data — a *Java*/*SQL* version and simplified pure *SQL* version. Main differences between the two implementations are:

- Runtime performance

  Implementing a tree search in *Java* and *SQL* leads to a very high communication overhead as each node expansion triggers an individual *SQL* query. A pure *SQL* implementation is therefore, unsurprisingly, substantially faster, which in particular becomes apparent when a runtime evaluation is done on synthetic data.

- Reliability

  The queries of the pure *SQL* implementation turned out to be too complex when confronted with huge amounts of data in the real world setting at hand. First experimentations with the pure *SQL* implementation repeatedly were abortive due to database overload. As the database management system had to be treated as an immutable blackbox, the definite causes and limits could not be determined. However, the highly variable workload and architecture optimization for other applications are assumed to have a considerable influence . The *Java* implementation with its numerous but simple individual queries and constant result logging proved to be more reliable and feasible under the given conditions with

massive amounts of data.

- Functionality / Expressivity

  The pure *SQL* implementation is very limited in functionality and expressivity, in particular as the use of extended constructs such as stored procedures or user defined functions was not possible.

- Configurability / Extensibility

  Likewise, the *Java* implementation is easier to configure and to extend which motivated the use of this version both in experimentation and discussion with business experts.

In summary, it can be stated that the *SQL*-Version was used for experimentation on limited subsets to produce fast results while the *Java/SQL* version was used whenever the higher expressivity and reliability was necessary, and optimized runtime was secondary.

3.2.2 *Excursus*: An Alternative Idea - The GraphSlider

The underlying idea of the *GraphSlider* is — instead of starting at a dedicated root node and expanding selected paths from there — to move a sliding window over the graph's time dimension. First, transactions (an according nodes) within the interval $[t_0, t_1]$ (where $t_1 = t_0 + WindowSize$) are loaded. At each time step, the sliding window is moved forward according to the configured step size — that is, transactions in the fringe $[t_1, t_1 + StepSize]$ are added, transactions in the fringe $[t_0, t_0 + StepSize]$ are discarded). At each time step, three functions are applied to the subgraph in the current time window:

- Scoring Function

  The scoring function assigns a score s to each node n_i depending on its direct neighbors $M_i \subset N$ and the transactions between n_i and its direct neighbors $T_i \subset E$, that is, ($s_{n_i} = f(M_i, T_i)$. One or more scoring functions accounting for different properties can be applied and added to a nodes score. Possible scoring functions are for instance:

  - Intermediary Node Scoring

    A nodes score is increased for each outgoing transaction matching an incoming transaction within the current time window in terms of its amount. The shorter two transactions succeed, the higher the resulting score will be due to repeated scoring after window propagation. A possible variant is the use of aggregated amounts within the time window for structuring detection. Aggregation may be done in account pairs or over all connected originators and beneficiaries .

  - Regularity Scoring

    A node's score is decreased if the transactions in the current time window are assessed to be regular. With the help of a regularity index, it is decided if the relation of two accounts is based on regular payments, which typically reduces suspiciousness. Numerous legal recurrent payment relations exhibit transactions in regular time intervals (e.g. monthly wage or interest payments).

- Spreading Function

  After scoring, a spreading function spreads a fraction f of a nodes score to its direct neighbors. This allows propagating scores over the network and accounting for connected sub-

graphs of high scoring nodes. The spreading of scores can be seen as a variant of the guilt-by-association-concept [Macskassy and Provost, 2005].

- Ageing Function (Temporal weighting)

  The ageing function decreases the score of a node at each time step right after propagation (prior to scoring and spreading). This accounts for the fact that recent events are of higher relevance for the current state than events that occurred earlier.

  If a node's score exceeds a defined threshold, the node is added to a persistent alert graph with its transactions in the current time window. The alert graph therefore represents a subgraph consisting of structures of interest.

  An illustrative version of the GraphSlider was implemented. The implementation was extended and evaluated in [Meier, 2009]. Although the approach showed some potential, the hardware resource requirements of the chosen implementation was prohibitive for an evaluation on real data. Even if performance problems could have been resolved, the problem remains that this approach needs extensive parameter setting. Finding the optimal settings requires time-consuming exploration. It soon became clear that this approach is too complex to experience quick wins and acceptance in Alphafin, why further development was abandoned. More detailed information on the GraphSlider and evaluation on synthetic data is available in [Meier, 2009].

3.2.3 The Transaction Matcher (*TMatch*)

TMatch is another implementation of the detection component which provides extended structural analysis. The structural analysis may be done on top of ChainFinder results and/or results from other pattern matching algorithms. Therefore, the ChainFinder algorithm may be seen as a possible, but not required part of TMatch. In the following, the considerations that motivate the development of TMatch are given. After this, the modules constituting TMatch are introduced and described.

Problem description

So far, we have focused on detecting relevant pattern instances in the data. After the first experiments with the ChainFinder and TVIS on real data, it soon became clear that single occurrences of those patterns may, at least at Alphafin, occasionally occur in non-fraudulent behaviour — which is not very surprising. Instead of looking at single pattern matches, it turned out to be of much higher interest to investigate groups of pattern matches that in some way belong together, for example all transaction chains triggered by the same employee. When looking at example visualizations, internal fraud experts confirmed that highly interconnected groups of pattern matches are much more interesting than disjointed single occurrences.

Figure 3.2.3 gives an illustrative example. In (a), a group of five identified chains for one employee is given in both timeline (left) and network view (right). The single occurrences of chains are not interconnected to each other. This leads to small, isolated components in the network view. In (b), the same views are shown for another employee. The chain transactions result in a more interlinked structure, which may be of higher interest[3].

[3] Such a structural analysis of the network view can be done using TVIS— a detailed real world example is given in case study I (4.3).

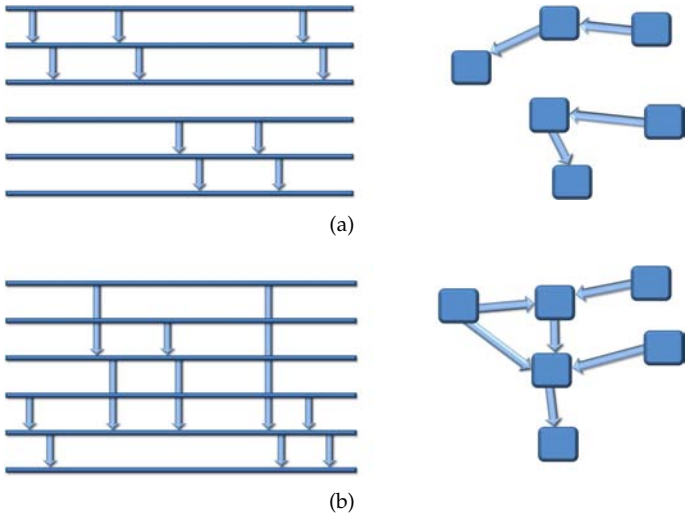


Figure 3.8: Two examples of chain groups

When relying solely on the ChainFinder in the detection component, structural properties of graphs consisting of grouped chains are assessed manually. TMatch, as a consequence, aims at automating the structural analysis of groups of pattern matches by finding and scoring graph structures of interest.

Solution Approach: TMatch Functionality Overview

TMatch works as follows: Given a graph (which may consist of pattern matches), it identifies connected subgraphs of a minimal user-defined size and assigns each subgraph a number of scores according to its structural attributes. Similar to the concept of "centerpiece subgraphs" [Tong and Faloutsos, 2006], the user is allowed to define a number of nodes (the *StartSet*) which serve as starting points in the search for components. This is in particular relevant for AML related topics where the definition of the population and the prior pattern matchers may be less restrictive as in internal fraud, which leads to very large graphs[4].

For example, the *StartSet* may consist of customers of a defined region that exhibit uncommonly high cash transaction activity (*cash customers*). The total population analyzed may be all

[4]If the analyzed graph is relatively small as in typical *IF* investigations, the limiting factor of a dedicated *StartSet* and the according incremental search may not make sense. Instead, the whole graph is typically loaded into memory and searched for connected subgraphs.

the customers in the region of interest. The *StartSet* is therefore a subset of the population. The goal is to identify all the connected subgraphs that contain at least one of the *cash customers*. The score of a subgraph is typically the higher the more *cash customers* it contains.

More generally speaking, the goal is to find clusters of customers that are of particular interest. As in [Tong et al., 2007], connections between StartSet nodes via other nodes in the population are found with this strategy. Figure 3.9 illustrates example structures identified by TMatch[5]. Nodes in the StartSet are marked red. The structure encircled green is a structure that is identified by TMatch as it contains at least one node from the StartSet . The red encircled structure is ignored as it only consists of nodes that are not in the StartSet . The other subgraphs are not retrieved as the minimal size (in this case 4) is not reached. Similar constraints can also be added concerning for example the minimal graph density within a connected subgraph.

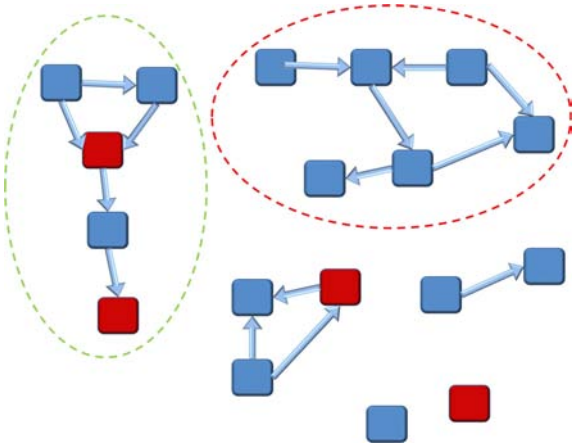


Figure 3.9: TMatch graph component identification example

TMatch Modules

By providing strucutral scoring of groups of pattern matches, *TMatch* constitutes an extended detection component of which the ChainFinder is (or can be) a part of. The overall architecture of *TMatch* consists of the modules illustrated in figure 3.10. In the following, those modules are shortly discussed.

[5] In graph theory, subgraphs are also called *(graph) components*. We will use these two terms interchangeably in the following.

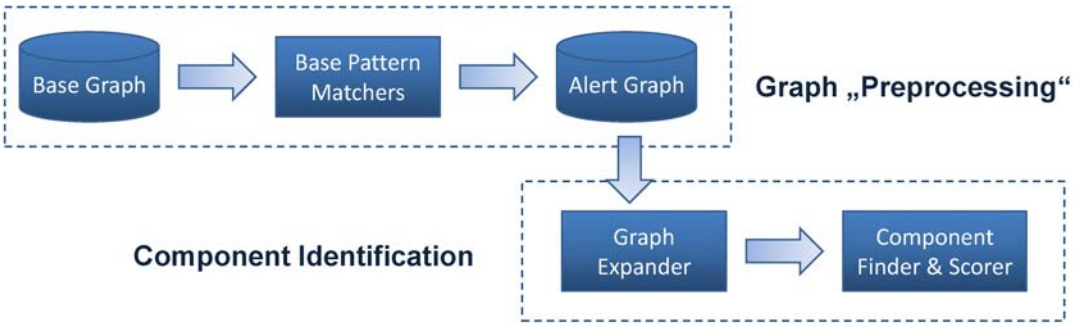


Figure 3.10: The TMatch modules

- Base Graph

  The Base Graph is implemented as a database view. It's function is to reformat the underlying data and present it in a suitable way to the subsequent modules. The Base Graph may represent a partition of the entire available graph, typically in terms of selected business regions. For performance and feasibility reasons, this view should be kept relatively simple. In settings where the database provides the data in a suitable form, it may be possible to do without a view and directly access the data tables. In Alphafin, however, the complexity of the database structure required a comprehensive optimized view definition.

- Base Pattern Matchers

  One or several Base Pattern Matchers may be applied to the Base Graph, condensing it to structures of special interest . The ChainFinder is an example of a Base Pattern Matcher. Matches are transferred to an *Alert Graph* for further analysis. A simpler, but for AML topics valuable example of a Base Pattern Matcher is the "High Amount Path"-Matcher. It only transfers edges exceeding a minimal aggregated transaction amount to the Alert Graph[6]. The Alert Graph, as the one introduced in the GraphSlider, consequently solely consists of patterns of special interest[7].

[6]Note that the ChainFinder, given according particle compatibility definitions, accomplishes this goal.

[7]Defining the Base Graph and applying the Base Pattern Matchers to form the Alert Graph could theoretically be combined into one, more complex database view. Separating the two steps however proved to be more flexible and increased reusability in application.

- Alert Graph

  As described above, the Alert Graph combines the results, or generated alerts of the Base Pattern Matchers. Combination of results may be conjunctive or disjunctive. In our experiments, we typically restricted the number of Base Pattern Matchers to one, to keep the end results as interpretable as possible. If no Base Pattern Matcher is applied, the Alert Graph is represented by the entire Base Graph.

- Graph Expander

  The basic idea of the Graph Expander is motivated by business requirements. The expander module starts analysis of the given graph at a number of user-defined nodes, which form the StartSet. From each StartSet node, the graph is expanded, recursively looking for adjacent nodes. Whether an adjacent node is considered or ignored depends on the configuration settings of TMatch. For example, only nodes with a minimal turnover or an in/outflow ratio similar to 1 may be considered.

  The expansion process is highly configurable. Expansion of a single path is closed when no connections are left or the search depth reaches a configured maximum. To avoid redundant work, path expansion also finishes when another member of the *StartSet* is met (as a new expansion will start from this point) or a node already visited is detected.

  Several implementations are available to optimize the runtime/performance tradeoff according to the size of the graph and resources at hand. In particular, a RAM based implementation can be used for small graph partitions, while a DB version is available, which incrementally loads the required data and keeps the memory used at a minimum.This allows for the analysis of huge datasets. A threaded version was implemented to optimize runtime in the presence of a resilient data base system.

- (Graph) Component Finder

  The Graph Component Finder uses the results from the previous expander step and calculates all the maximal connected components. After this, the Component Finder filters the graph components according to the configuration, calculates additional metrics, and stores the remaining subgraphs and metrics in the database. The result is a list of subgraphs, ordered by node size. Table 3.1 describes the used metrics.

Metric	Summary
Graph Density	The graph (degree) density is calculated as the number of edges |E| divided by the possible number of edges (|N|(|N|-1) in a directed graph) [Wasserman and Faust, 1994]
Degree Centrality	The degree centrality of a node is defined as as the number of links incident upon the node divided by the possible number of links (|N|-1) [Wasserman and Faust, 1994]. In the basic implementation, we calculate the undirected degree centrality. The maximal degree centrality occurring in a subgraph is logged. A value of 1 for a node means that the node is connected with all other nodes in the subgraph. A maximum centrality of 1 and a low graph density in a subgraph are indicators of a "star structure".
Cycle-Node-Ratio	The cycle-node-ratio is calculated by creating a subgraph of all cycles. The ratio is the node size of the found cycle-subgraph, divided by the size of the analyzed subgraph from the Component Finder results.
Total-Amount-In	This value contains the sum of all incoming transaction amounts.
Total-Amount-Out	This value contains the sum of all outgoing transaction amounts.
Total-Amount-In Flag	If the Total-Amount-In value is higher or equals than the configured threshold, this flag will be set to 1.
Total-Amount-Out Flag	If the Total-Amount-Out value is higher or equals than the configured threshold, this flag will be set to 1.
Passage Flag	If the difference of Total-Amount-In and Total-Amount-Out is higher or equals than the configured threshold, this flag will be set to 1.

Table 3.1: The calculated component metrics and flags.

- Component scorer

  After separation into graph components, several scorer models are loaded for each subgraph, calculate the score and save it to the database. Individual scorers can be added or removed in configuration. The scorer models currently included are:

  - PassageScore

    denotes the fraction of passage nodes in the component $\frac{|P_i|}{|N_i|}$ where P_i is the set of passage nodes in component i and N_i is the set of all nodes in component i. If a node is a passage node is calculated in the *Component Finder*.

  - PassagePathScore

    The set of *PassagePaths* X_i in component i is defined by specified transaction paths[8] in component i where every node n_i in the path is an element of P_i. Each PassagePath $x \in X_i$ has a maximal length depending on the number of visited passage nodes[9]. The PassagePathScore represents the average length of all PassagePaths j in component i $\frac{\sum length(x)}{|X_i|}$.

  - TransactionTypeCountScore

    calculates the fraction of transaction of a certain type $\frac{|E_t|}{|E_i|}$ where E_t is the set of edges of $1 - n$ transaction types in a user defined list in component i and E_i is the set of all transaction in component i.

  - TransactionTypeAmountScore

    Instead of counting occurrences, this scorer calculates the sum of transaction amounts in E_t and its fraction of the sum of all transaction amounts in E_i, that is,

    $$\frac{\sum_{y \in E_t} amount(y)}{\sum_{z \in E_i} amount(z)}$$

  - MaximumFlowScore

    This scorer makes use of an additional user defined set N_{sink} which denotes possible target or sink accounts in the analyzed population, for instance accounts with a

[8] for example, transaction chains
[9] note that the ChainFinder algorithm is designed to only log maximal length paths

considerable cash outflow. The maximum flow between nodes $n^i_{start} \in (N_{start} \cap N_i)$ and nodes $n^i_{sink} \in (N_{sink} \cap N_i)$ is calculated. This is done by adding two artificial *meta-nodes* to component i, one connecting all nodes in N^i_{start} and one connecting all nodes in N^i_{sink}. After this, the *Edmonds-Karp Maximum Flow algorithm* [Edmonds and Karp, 1972] is used to calculate the maximum flow between these two *meta-nodes* in component i.

– StartSetScoreModel

This score calculates the fraction of all *start nodes* in component i $\frac{|N^i_{start}|}{|N_i|}$.

– SizeScore

is represented by $\frac{|N_i|}{\sum_{i=0}^{n}|N_i|}$ and denotes the size of component i according to all nodes in the analyzed population being part of any component.

- TMatchViz

TMatchViz is a basic graphical user interface for TMatch, providing access to predefined search settings and defining new search configurations. Identified components and their scores are displayed in a result view. TMatchViz also offers a very basic graph visualization for quick assessment of components. Detailed investigation of results is typically done in the visualization component TVIS.

A more detailed descriptions of architecture and implementation issues of TMatch can be found in [Moll, 2009].

4

Application and Evaluation

In this section, a number of evaluation issues are considered. How can such a system be evaluated in a meaningful way? How were similar systems and approaches evaluated? These are the questions we try answer. After this, two case studies describing real world applications are given as detailed as non-disclosure agreements allow. For completeness, a synthetic data evaluation is given and the problem of synthetic generation is discussed in a nutshell. As stated above, an extensive evaluation of the visualization component in ad hoc use has not been made. However, discussions with several potential future business stakeholders suggest an informal cost savings estimation in comparison to traditional, table based analysis of data, which concludes the application chapter.

4.1 Application Considerations

As mentioned in the related work section, numerous analytical fraud detection approaches in research rely on labeled data, that is, on a reasonable number of both positive and negative examples. As Bruce Schneier puts it, "Data Mining works best when there's a well-defined profile you're searching for, a reasonable number of attacks per year, and a low cost of false alarms."[Schneier, 2006] This is not only an ideal initial situation for the application of a large number of ready-to-use supervised data mining algorithms, but also makes evaluation straightforward. The performance of the detection component can be given by means of a *confusion matrix* or a *ROC-Curve* and is well defined and directly comparable to other models which are calculated

on the same data.

As soon as labels are not available — as in our case —, evaluation is more problematic and a number of issues has to be considered that we introduced in the first chapter (Section 1.2) and review at this point in a nutshell.

Experts are needed to assess the results and decide if the classification of an instance or pattern as fraudulent (or, in a weaker form, as ”relevant” or ”interesting”) by the detection component is justified or a false alert. The true class of a pattern may not be straightforward to assign even for an expert and might also be a matter of opinion.

While the ratio of false alerts, also called *false positives* may be approximately estimated this way, the problem of a completely unknown false negative rate remains. It lies in the nature of the problem that there always may be fraudulent instances or patterns in the data which nobody (except the fraudster) has knowledge of.

Detection components, if not relying solely on identification knowledge (labels), require human expert input. The overall assumptions, the corresponding choice and design as well as the detailed configuration of a detection approach may (and should) be influenced by the information available from human experts. This means that, in a certain sense, evaluation of a real world fraud detection system inseparably includes evaluation of human expert knowledge.

Furthermore, a fraud system improves with the growing experience of its users, possibly over years. First applications to real world problems may have an explorative character which, ideally, reveals potential. Apart from the quality of expert knowledge, performance of a real world fraud detection system may therewith rather measure the level of available experience and refinement during research than the performance of the system itself.

Concerning similar fraud detection systems, *LAW [Wolverton et al., 2003]* limits evaluation to matching runtime. For *FAIS* [Senator et al., 1995] and *ADS* [Kirkland et al., 1998] however, the authors are able to reports numbers that emerged from using the system for several years. While for *ADS* , ”hits” are defined as ”breaks that have resulted in follow-up actions[1]”, the article on *FAIS* reports, ”109 feedback forms from outside agencies in addition to feedback from inhouse investigations, where 90 percent of the feedback indicates either new cases opened or relevance to ongoing investigations” without further explanation. Additionally, one closed case with follow-

[1] which is, as reported, the case for 800 out of 7000 breaks during 9 months

up investigation, prosecution and conviction resulting from a lead generated by the system is reported[2].

Unfortunately, we are not able to deliver such numbers as the productive use for years and the "dedicated group of intelligence analysts engaged full-time in reviewing, validating and pursuing potential leads generated by the system" as for example mentioned in [Senator et al., 1995] is not given.

As our approach heavily relies on graph pattern matching, common evaluation in this research area may be considered as well. The survey by Gallagher [Gallagher, 2006a] discusses the topic of graph pattern matching evaluation. Most publications deliver runtime performance measures. As graph properties and algorithm designs and requirements are highly varying between different publications and fields of application, comparing runtime is of very limited significance. Furthermore, while an optimization of runtime may be the main issue from a theoretic point of view in the emerging graph pattern matching research area, feasibility is the main concern in applied research. As we learned when working with the *ChainFinder*, the fastest implementation is not always the most feasible one. At this point it is worth noting that all graph pattern matching approaches in the survey [Gallagher, 2006a] were only applied to substantially smaller graphs than the one our approaches were applied to.

The considerations above, given the small size of our project and the limited expert resources, led to the decision to evaluate our system by means of exemplary case studies on real world problems. As defined in the introduction, expert assessment in terms of interestingness was our main concern when evaluating. Other evaluation measures are partly used where justified by the case study and are explained in the according section.

[2]The authors also give the number of analyzed transactions (approximately 200'000 transactions each week) and the number of leads for 1993 (27), 1994 (75) and part of 1995 (>300).

4.2 The Data Set

The data sets used for the following case studies are based on a data warehouse which has been developed at Alphafin for analytic purposes in the area of fraud, compliance and risk related topics.

The integrated view which became possible with the development of such a data warehouse forms a good initial situation. The complexity of the data required comprehensive business understanding and considerable data preprocessing even for the most basic views. Acting upon fraud expert advice, we therefore minimized the amount of data in terms of data fields for simplicity and feasibility reasons in the context of this thesis. The data warehouse contains all transactions within a time window of two years to the current date. The exact size of the corresponding graph is therefore constantly changing. The number of all transactions available is more than one billion, involving 16 million customer accounts and a considerably higher number of accounts external to the financial institute. To illustrate the nature of the transaction graph, Figure 4.1 on the facing page shows the in and out degrees of the network within Alphafin (without transactions across company borders).

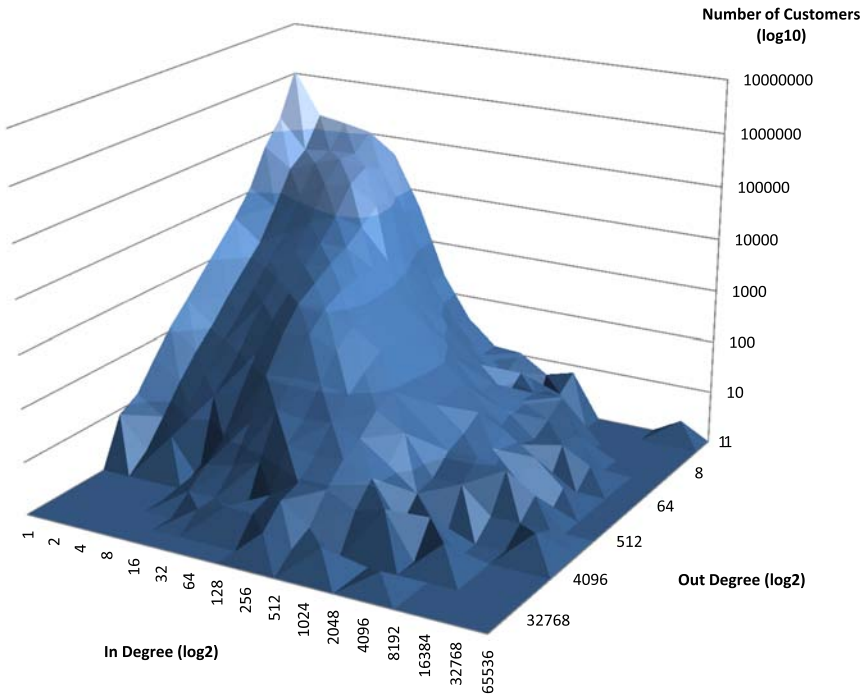


Figure 4.1: Real world data degree distribution between customers of Alphafin

4.3 Case Study I: Internal Fraud Analysis

In collaboration with fraud experts, an example case study on real world data was defined and conducted. The results presented in this thesis had to be made anonymous for confidentiality reasons. Accounts and customers respectively are labeled with identification information in TVIS, but appear blank in the following figures. As the tabular view cannot be given, but is crucial for the assessment of structures, we try to partially compensate the information loss with more general and therefore noncritical explanations. In addition, it has to be stated that a detailed investigation includes further data sources and systems — as for example the client contact history.

4.3.1 Evaluation Goals

The goal of this case study was to evaluate the following three aspects:

- Commonness of Chain Structures

  Although transaction chains are known to play a major role in numerous analyzed fraud cases, the commonness of these structures in normal transaction behavior was widely unknown. By examining the frequency and context of transaction chains, their discriminative power in fraud detection may be estimated. How common are chains? Are there defined line-ups where chains emerge in normal, daily business? These and similar question were considered in the case study.

- Identification of a Reference Fraud Case

  In particular, a reference case contained in the available data was proposed by fraud experts. The ChainFinder was configured without detailed knowledge of the case but common model knowledge. The ranking of the reference case in the ChainFinders suspiciousness scoring model (precision) and the portion of detected transactions being part of the case (recall) was evaluated.

- Investigation Efficiency

  This aspect evaluates the cooperation of the ChainFinder with TVIS. The following questions were relevant: What is the average workload of investigating the produced alerts, in particular in comparison to existing fraud detection methods? May the focus on relational

transaction structures instead of isolated transactions in combination with visualization improve investigation efficiency?

4.3.2 Data Basis and Configuration

The entire available dataset, representing a time window of two years, was scanned for chains of a dedicated transaction type, *Manual Transaction Type* (*MTT*), which we introduced and described in section 2.3.1.

A this point, we will repeat the characteristics of *MTT* transactions relevant for internal fraud detection. Employees can trigger *MTT* transactions autonomously, that is, without customer order in written form. While being a valuable tool for unbureaucratic and flexible customer service, *MTT* is prone to misuse. Above a defined threshold *th*, additional control measures kick in. Therefore, transactions with an amount $> th$ were excluded from this analysis. Below this threshold, fraud detection performs the task to impede misuse without decreasing the flexibility and value of the service. The overall number of distinct customers in the analyzed data set was approximately 16 millions. For this case study, the root set was defined by each customer featuring one or more outgoing *MTT* transactions. This definition is very general and was chosen with the intention to separate the evaluation of chain structures from other known model knowledge implemented in existing fraud filters. As the number of relevant *MTT* transactions was 1.1 millions, a large part of customers did not exhibit any *MTT* transaction and could be excluded for the initial chain finding. During the investigation of the results in the Transaction Visualizer, additional transaction types were occasionally loaded, increasing the number of analyzed customers in the whole process.

Furthermore, the scope of the search was limited to transactions between Alphafin clients (no external transactions). However, chains leaving the bank (e.g. ending in a cash transaction or a *MTT* transaction to an external account) were retrieved using a second *ChainFinder* run where the initial structure based on internal chains suggested it. This approach appeared to be the best trade-off between performance and effectiveness for the rapid analysis of the entire data set given the limited resources for our study. A more focused search may incorporate external transactions from start. The most important configuration settings for the *ChainFinder* were defined as follows:

- Single chains of *MTT* transactions with an amount $< th$

- Registrator tracking enabled: single transactions forming a chain are registered by the same employee.
- Money remains at most seven days on intermediary accounts.
- Amount may differ within a chain up to 15%.

A simple scoring model based on the number of registered transaction chains was applied. A higher number of registered *MTT* transaction chains resulted in a higher interestingness score. For this case study, employees with a minimum of 5 registered chains were proposed for further analysis.

It has to be repeated that techniques involving other transaction types (e.g., document or signature forgery) may also be used in internal fraud as good as in external fraud, possibly leading to chains and smurfing structures in a wide variety of transaction types. However, the inhibition threshold for these actions may be higher than for misusing *MTT* transactions, which are just one click away. The limitation to *MTT* transactions may therefore be reasonable for a starting point.

4.3.3 Commonness of Chain Structures

The overall results indicate the frequency of chain structures for *MTT* transactions: Figure 4.3 reveals that, before any application of exclusion criteria (see below), less than 2% of the *MTT* transactions in question are part of transaction chains.

First insights motivated the definition of a number of exclusion criteria, which further reduced identified chains and led to a ratio of less than 1%. The average chain length (that is, involved transactions) is approximately 1.8. This fact requires explanation. While the better part of the chains are of length 2, one transaction, in particular in high activity context, can repeatedly be accounted for multiple chains due to particle cloning. Figure 4.2 illustrates this effect.

On the other hand, transaction chains up to length 6 were observed. The ratio of employees with at least 5 *MTT*-chains to all relevant employees which registered at least one *MTT* transaction is similar. From 10053 relevant employees, 100 were proposed for further analysis by the ChainFinder, leading to a ratio of 1%. Figure 4.4 shows the distribution of those employees according to the number of registered chains.

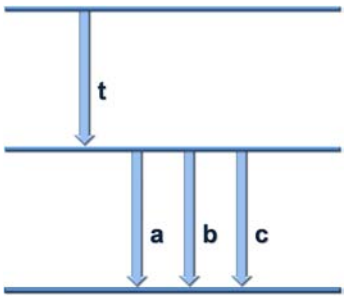


Figure 4.2: A particle coming along transaction t will be cloned three times to travel along a,b, and c. This results in 4 transactions that form 3 chains of length 2.

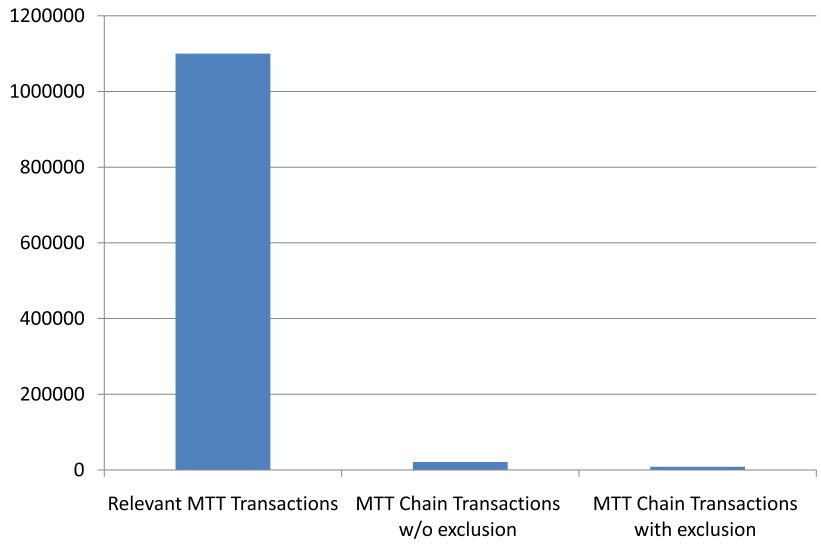


Figure 4.3: Total number of relevant transactions and number of transactions in chains

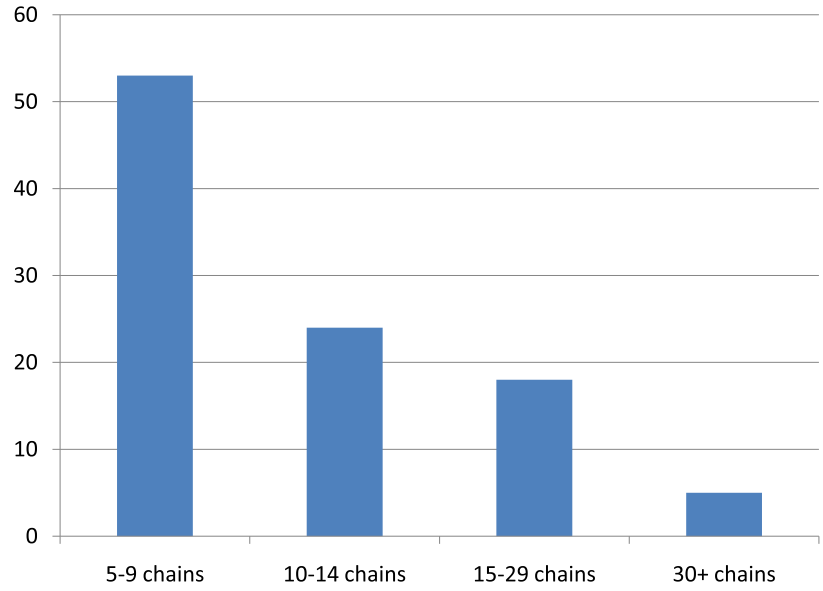


Figure 4.4: Number of employees and registered chains

These results indicate that for the defined transactions of interest, transaction chains are rare. Of course, as long as it is unknown if any of the proposed employee is actually fraudulent, a statement on the discriminative power of chain structures in terms of fraud detection cannot be made. However, the low number of produced alerts makes investigation feasible and suggests potential under the observation that transaction chains are repeatedly present in analyzed fraud cases.

Results were visualized and a selection was discussed with internal fraud experts. This led to the identification of previously unknown settings that produced a high number of chains but did not appear to emerge from fraudulent behavior. An example if given in Fig 4.5 . The star structure consists of an asset management company in the center and its customers. On two key dates, a massive amount of transactions is triggered from and two customer accounts. It remains unclear if any of the occurring chains are intended as such, but given the massive activity on two single days with transactions in a limited amount range, the emergence of numerous chains structures is inevitable. The equally high number of incoming and outgoing transactions of the center node and the wide variety of amounts led to a high number of matches in spite of the random structure

Score (= number of chains)	Employee (anonymized)
241	A
54	B
40	C
38	**D**
30	E
29	F
27	G
27	H
24	I
22	J

Table 4.1: The ten top scoring structures

identification in this special case. While the structure was considered to be explicable, the excessive use of *MTT* transactions in this setting was confirmed to be very unusual and suggested for investigation at the compliance department. Although the transaction volume of this example was unique, similar, but smaller structures were repeatedly found, which suggested the definition of according exclusion criteria. This example illustrates how the identification and visualization of patterns can lead to new, structure-based knowledge of the data which may be of use in a wide variety of monitoring applications.

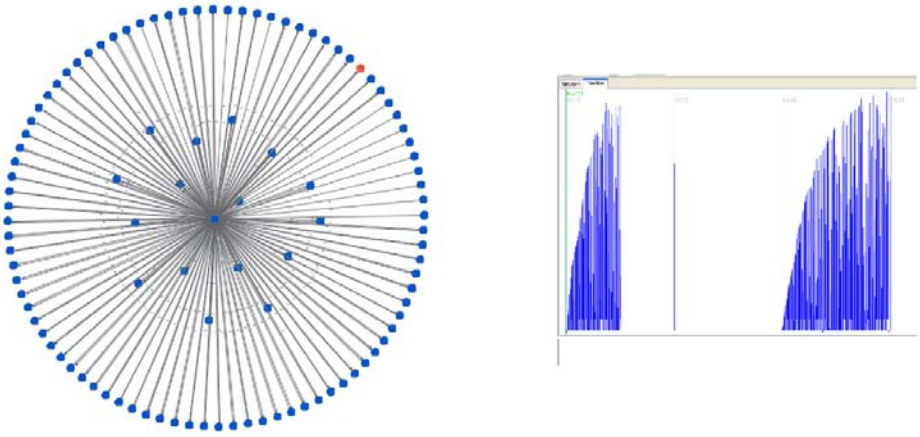


Figure 4.5: A chain structure of limited interest

4.3.4 Identification of a Reference Fraud Case

Table 4.1 shows the ten top scoring structures in this case study.
Employee *D* turned out to be the perpetrator of the reference fraud case. The fact that the reference case is located in the "Top 5" of fraud scores on the overall data set may indicate a good precision. A discussion of this statement is given below. The evaluation also showed that the ChainFinder retrieved a substantially higher number of the transactions in the case in comparison with existing detection methods. With existing SQL-Filters, 4 transactions from this case triggered alerts. Manual analysis then showed that the whole fraud case actually consisted of more than 100 fraudulent transactions. In contrast, *ChainFinder* at first successfully identified approximately 60% of the fraudulent transactions and more than 90% after conducting the additional external chain search. It may be assumed that, had the delinquent been only slightly more careful, the existing *SQL*-Filters wouldn't have detected this quite extensive case but nevertheless it would have been found by the *ChainFinder* — but this, of course, remains speculation.

It may be argued that the detection of a case basically consisting of chain structures is not a big accomplishment given the *ChainFinder* algorithm. We fully agree on this statement. However, as we decided on a pattern matching approach (as motivated in chapter 2), our intention is not the identification of previously completely unknown structures, but to examine the potential of relational model knowledge in addition to the non-relational model knowledge in existing fraud detection monitors at Alphafin. The fact that our algorithm was able to find the reference case is not surprising and goes without saying. The fact that it ranks among the Top 5 scoring structures even with a trivial scoring mechanism (solely relying on the number of chains identified) however shows the discriminative potential of this approach, which was previously unknown.

4.3.5 Evaluation Efficiency

All the chain structures which were proposed for investigation by the *ChainFinder* could be assessed in approximately three working days. This basic first assessment included the classification of each structure into low, medium and high interestingness level. Chain structures of low interestingness essentially consisted of numerous isolated transaction pairs and triplets which exhibited plausible reasons for the chain transactions. It has to be stated that this first assessment

was not conducted by fraud experts at Alphafin due to the lack of resources, but rather served as a preselection to reduce the number of patterns presented to experts to a minimum. This resulted in 4 structures which were confirmed to be of particular interest by experts. A closer investigation of these structures was reported to be intended, but never happened to our knowledge. The following section illustrates how the high scoring employee behaviors were analyzed by means of two examples.

4.3.6 Example Investigation

This example demonstrates the investigation of both employee B, which reached a top score, and employee A, which exhibits a known fraud case. It shows how visualization either intensifies or weakens suspiciousness. For investigation, the structures resulting of all found MTT chains of employee B and employee A, respectively, were visualized in TVIS, which could be done within a few seconds (figure 4.6)

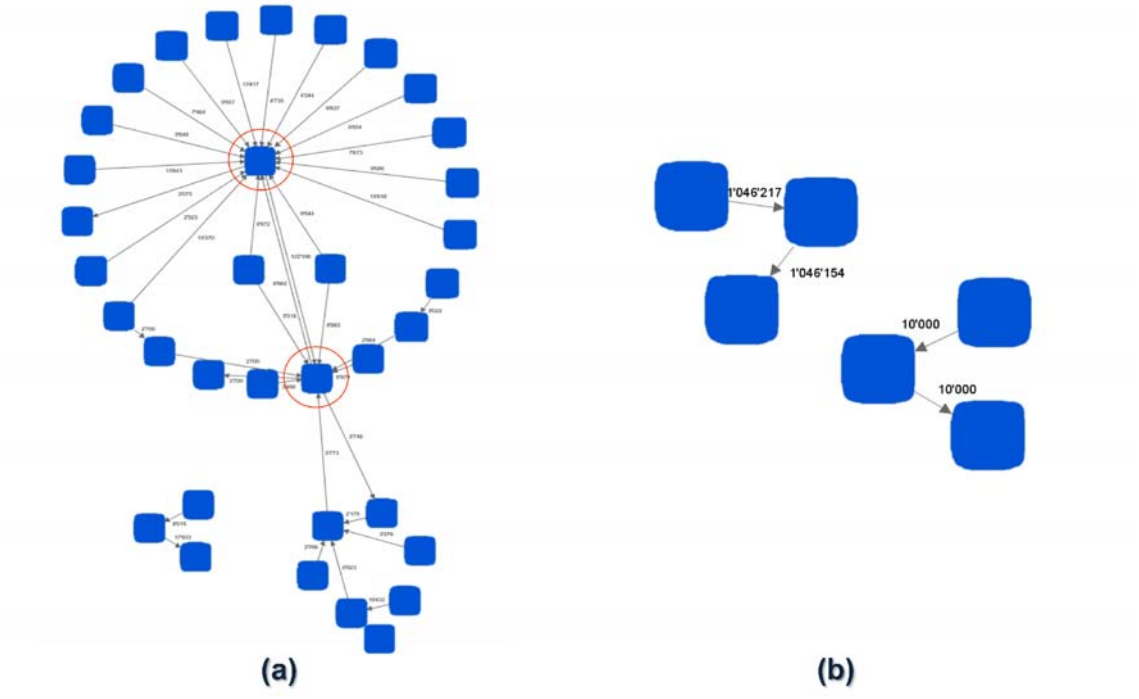


Figure 4.6: Chain structures for registrator A (a) and B (b). Accounts serving as accumulative targets are marked red.

The visualization shows that although employee A has substantially less chains (the total number of logged chains is 38) than B (241 chains), his/her structure looks way more complex in the network view. Within the structure, two accounts seem to operate as accumulative targets (marked red), while numerous accounts occur as originators only. In the tabular data view in TVIS (not visible in the Figure) it is straightforward to see that most of the originator accounts are numbered accounts, which intensifies suspiciousness according to experts. In contrast, the huge number of chains that B registered only show up in a structure of two simple triplets. This, of course, doesn't necessarily lead to the conclusion that this structure is not suspicious, particularly because the originators and intermediaries are also numbered accounts[3].

To get more insights, we make use of the dynamic visualization features of TVIS and add all *MTT* transactions — for the investigated customers and triggered by the employee in question — that are not part of a chain, but remain below the defined threshold. The idea behind this is straightforward: Extending the visualization this way reveals the general use of *MTT* transactions that are small enough to escape operational control mechanisms by the according employee. This may lead to possible conclusions about suspiciousness. Consider Figure 4.7:

The structure of the graph for A adds to the impression that money gets collected at (now three) accumulative targets and from there gets transported to one or more external accounts. The red node in the graph stands for 1-n external accounts (which were added in the extension step) and cash-out transactions — basically money leaving the bank.

The graph for B shows a high *MTT* activity on one of the originator accounts from the underlying structure. It may be of interest to further investigate if the high activity of this originator is appropriate; however, the graph does not contain any "money flow direction" suggesting suspicious transfers.

More information can be gained if — again just for the investigated customers — both *MTT* transactions *above* the defined threshold **and** *MTT* transactions registered by *other client advisors*[4] are visualised. The assumption is the following: If a "transaction path" is only used by one employee, which only triggers *MTT* transactions below threshold *th*, the probability of a "fraudulent path" is higher than if other employees also use this "path" and transactions higher than threshold *th*

[3] A numbered account is identified only by an arbitrary number instead of personal information for increased anonymity of the account holder.

[4] While a customer is typically assigned to one, dedicated customer advisor, it is perfectly common that other customer advisors occasionally trigger transactions substitutionally.

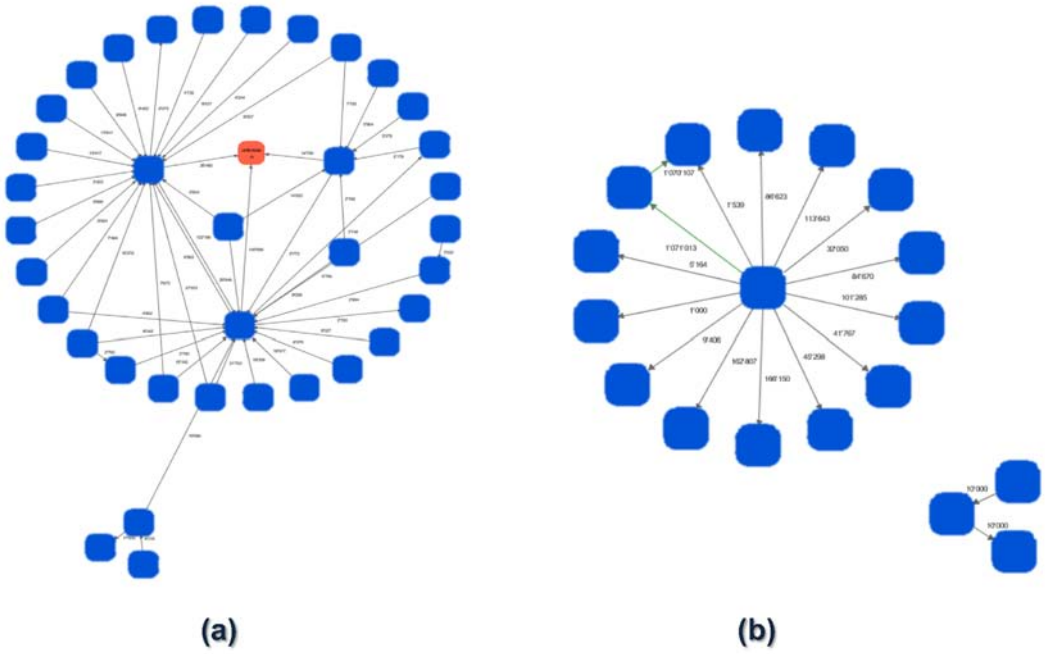


Figure 4.7: All "small" *MTT* transactions of the registrators in question

occur at least occasionally. For example, *MTT* transactions originating at a numbered account belonging to a high-age person may indicate the fraudulent evacuation of money if triggered exclusively by one employee and always staying below the critical threshold. The same transactions triggered by several client advisors or above the threshold result in certain control mechanisms to become effective and are therefore less suspicious. If we limit the visualization to transactions registered "by others" (or above threshold *th*), the figures change dramatically (figure 4.8)

While it is obvious that the graph on the right side stays similar (and therefore the transactions are most probably founded in customer orders and carried out by different client advisors), the graph on the left side changes substantially. Only few transaction in "this environment" are registered by others or above threshold *th*. These visualizations clearly add to the suspectedness of employee A while lowering the "interestingness" of employee B.

The manual work in the described investigation process motivates further automation. This consideration led to the development of TMatch, which is evaluated in the following case studies — however, with the focus on AML related topics.

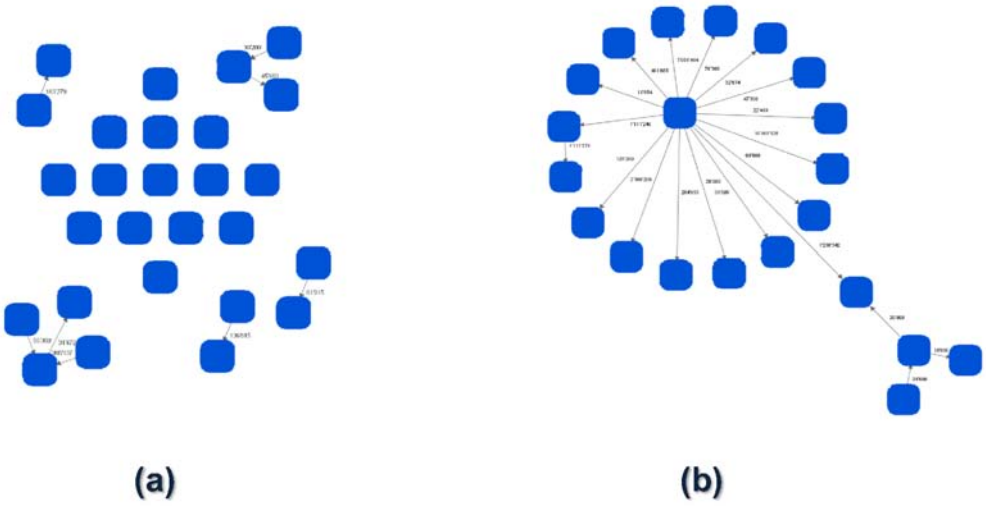


Figure 4.8: The same accounts for transactions registered by others

4.4 Case Studies II and III: AML/ Compliance Analysis

The combination of *TMatch* and TVIS was applied to two example AML-related problems.
In this thesis, the problem of money laundering merely serves as an alternative experimental application area. We did not conduct an extensive analysis of the field, but solely relied on expert input and on our basic understanding of the existing commercial AML solution at Alphafin. Published research dedicated to analytical money laundering detection seems to be relatively rare [Senator et al., 1995; Zhang et al., 2003; Wang and Yang, 2007]. Celent published a report on available commercial money laundering solutions in 2006 [Katkov, 2006]. As we only scratched the surface of this huge topic, both case studies are highly explorative, and findings remain vague.
While one of the problem definitions was open and general, the second one was more focused on a particular issue. As in case study I, the presented results are made anonymous and only very limited detail information can be given.
Both experts and responsible customer advisors lack experience in assessing and investigating relational structures. Therefore, the goal of this case study was to evaluate the following limited aspects:

- Number of results produced

Similar to the setting in the internal fraud study, it was completely unknown if TMatch would be able to produce a reasonable number of results. Even if precision and recall estimations may be very vague at this explorative stage , an adequate uncommonness of identified structures is a manifest requirement for producing results of interest for experts. An approach either producing no results or, less restrictively configured, retrieving the large part of analyzed accounts would not be able to fulfill this requirement.

- Expert assessment of interestingness

  As a detailed investigation and terminal assessment of the legality of structures was out of scope by far, informal expert assessments of interestingness and value were used to evaluate the quality of sample results. At this point it has to be repeated that parameter settings, which were discussed with experts, obviously influence the results produced. While evaluation of parameter setting and the approach as such is therefore inseparably combined, experts were aware of this fact in their assessments.

- Estimated cost savings

  Experts were asked to roughly estimate potential of TVIS and TMatch for efficiency and effectiveness improvements in comparison to current tools and techniques.

The organization of the following two AML-related case studies is as follows: First, a short problem description, quantitative results and selected retrieved structures are presented in a nutshell for both settings individually. After this, we consider the combined results of the two case studies according to the issues mentioned above. Possible implications of quantitative results are identified and the statements that resulted from discussing the structures with experts are given. Concluding, added value in terms of efficiency and effectiveness is considered. The high level and tentativeness of the expert statements is due to the fact that a more detailed investigation is a very elaborate undertaking including both compliance experts and responsible customer advisors, which was infeasible for this project.

4.4.1 Case Study II

Problem Description

This case study was conducted to evaluate the potential of supporting the processing of an audit issue, which was raised for a dedicated business region. The goal was to gain general insights and evaluate if our approach retrieves structure of interest for AML experts.
The analyzed region consisted of approximately 81700 accounts and 513000 transactions. Transactions to and from external accounts were not used for the identification of the structures in TMatch, but added to the visualizations. Exclusion from the structure identification is justified by the fact that the quality of available data does not allow to identify external accounts uniquely for each external transaction.
Two base pattern matchers were applied:

- A High Value Path pattern matcher which added each connection between nodes exceeding and aggregated amount of 1 Mio CHF to the alert graph
- A Sum Chain Pattern matcher (based on the ChainFinder) which added Sum Chains exceeding an aggregate value of 10000 CHF to the alert graph

Weakly and stronlgy components were identified and logged separately[5] .

Results

The following number of components were identified for the different settings:

High value path components (strongly connected) with Size $\geq$ 4: 10 Components were identified. Table 4.2 shows the number of identified components and their size. Only 0.06% of all analyzed accounts are part of a high value path component.

Figure 4.9 shows the biggest component of size 13. The size and density of this component is very unusual and was confirmed to be of high interest for AML and Compliance issues.The account marked red is a numbered account. The intense interconnections may indicate that the

[5] A component is strongly connected if for each pair of nodes n_i,n_j in the component there is a path from n_i to n_j, and a path from n_j to n_i where the path from n_i to n_j may contain different nodes and edges than the path from n_j to n_i (see [Wasserman and Faust, 1994] In contrast to [Wasserman and Faust, 1994] we denote each connected component which is not *strongly* connected as *weakly* connected for simplicity

Numer of components	Size
1	13
1	7
1	6
1	5
6	4
Number of components: 10	Number of nodes within components 55

Table 4.2: Strongly connected high value path components

involved accounts have the same beneficial owner. While this is not necessarily indicating suspicious activity, it is of interest whether the assumptions this pattern suggests match the knowledge and KYC[6] documentation of the responsible customer advisors. If the observed behaviour cannot be justified with the available information in the client history, the KYC principle may be violated and a detailed investigation is necessary.

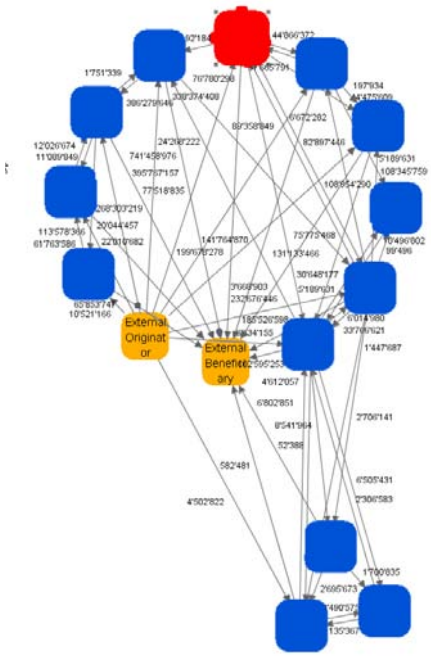


Figure 4.9: A component of high interest containing a numbered account (marked red)

[6] "Know Your Customer"

Numer of components	Size
1	254
1	24
1	13
2	11
1	10
Number of components: 6	Number of nodes within components: 323

Table 4.3: Weakly connected high value path components

High value path components (weakly connected) with Size $\geq$ 10: Due to the lower restrictions of weakly connected component, the minimal size was set to 10 nodes. Table 4.3 shows the number of identified components and their size.

The very large size of one component proposes experimenting with more restrictive high value paths in future. Extensive expert investigation, which was out of scope in this case study, may help to find the most suitable parameter settings. Figure 4.10 gives an example of a weak component of size 11. The component visualization was enriched with edges below the threshold level of one million for investigation.

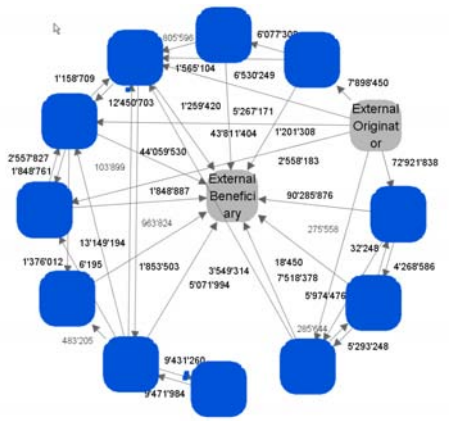


Figure 4.10: Example weakly connected high value path component

Sum chain components (strongly connected): The use of an account as a mere passage point may result in sum chains. These pattern may have perfectly legal reasons, for example emerge from dedicated company structures or may have illegal motivation (in particular layering [Altenkirch, 2006]) The idea of this analysis, combining the ChainFinder and TMatch, was to evalu-

Numer of components	Size
1	16
1	6
1	5
8	3
Number of components: 11	Number of nodes within components: 51

Table 4.4: Strongly connected sum chain components

ate if complex structures consisting of Sum Chains exist, and if yes, how common they are. Table 4.4 shows the number of identified components and their size.

Figure 4.11 visualizes the biggest component of size 16. The star structure, which is indicated in the component scoring by a high maximum centrality (1), is of limited interest. The sum chains may have randomly occurred.

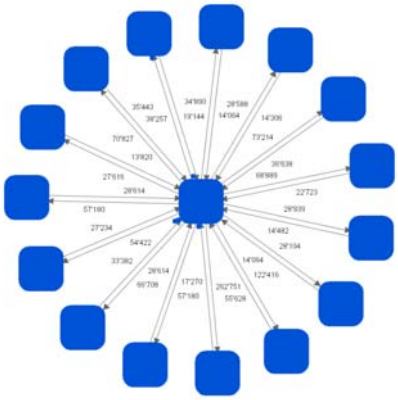


Figure 4.11: A strongly connected sum chain component of limited interest

In contrast, Figure 4.12 shows a strongly connected component (for internal nodes) which is of high interest. The structure exhibits numerous intersecting high value sum chains (simplified numbers), which is very uncommon.

Fig. 4.13 shows the same structure enriched with connections not being part of sum chains.

Sum chain components (weakly connected) with size ≥ 5 Table 4.5 shows the number of identified components and their size. The presence of a component of size 753 is striking. A component of this size is not suitable for investigation neither for TVIS in the current release nor for a human expert. The proposed way to evaluate such a component is therefore to run a more restrictively

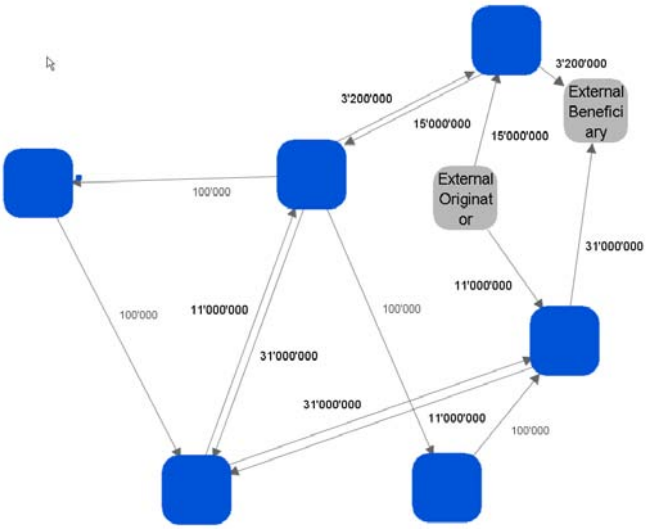


Figure 4.12: A strongly connected sum chain component

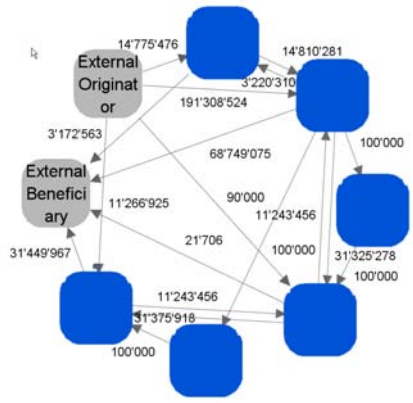


Figure 4.13: Enriched component visualization with all transactions

Numer of components	Size
1	735
1	16
1	13
1	8
6	7
4	6
4	5
Number of components 18	Number of nodes within components: 858

Table 4.5: Weakly connected sum chain components

configured instance of TMatch on this component and analyse the resulting subcomponents. The fact that one huge component repeatedly occurred in a number of settings (cf. Table 4.3 and 4.7), proposes the extension of TVIS to provide optimized visualization of big components (see chapter 5). The component of size 16 exhibits another complex structure consisting of sum chains (Figure 4.14).When the minimal amount for sum chains was raised to one million, 13 components remained (Table 4.6).

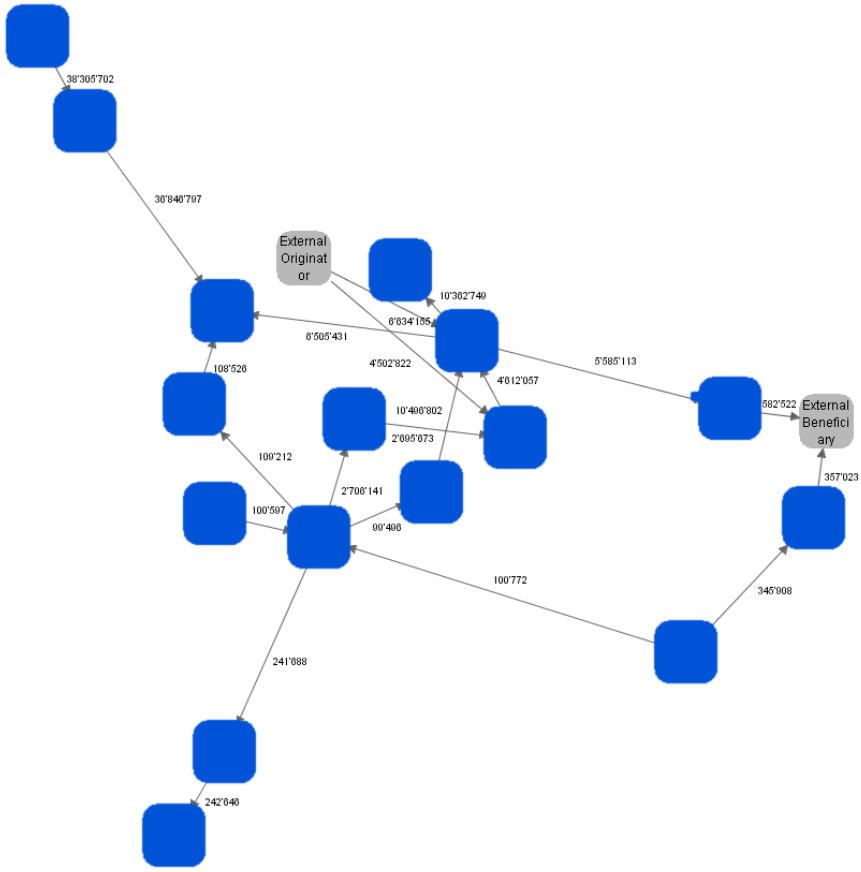


Figure 4.14: An example weak sum chain component

Numer of components	Size
2	10
2	7
4	6
5	5
Number of components 13	Number of nodes within components: 83

Table 4.6: Weakly connected sum chain components II

4.4.2 Case Study III

Problem description

This case study was conducted to evaluate the potential in a particular issue, which is described in this section.

The focus lies on a defined customer type we denote as *Uncommon Wealth Customers* (UWC). UWCs are registered as wealthy private customers in the bank and have according bank accounts. However, and this is what makes them uncommon, instead of using their accounts primarily for asset management as intended for this account type, UWCs exhibit massive business activity. This phenomenon is relevant in terms of AML, in particular for the *Know Your Customer Principle* (KYC). It has been observed that UWCs may operate complex structures of private accounts, representing company conglomerates. As this happens in an improper client segment, the responsible customer advisors often do not have sufficient knowledge of these behaviors and the underlying structures. Different parts of a company conglomerate making use of UWC accounts may be ministered by different customer advisors at different organizational units, which can have the effect that no one sees the whole picture. A project has been launched at Alphafin to gain more knowledge of this phenomenon. UWCs, or, more precisely UWC candidates are identified by defining SQL-Filters searching for accordant customer profiles. Currently, structures of interest are found manually on the basis identified UWCs. For example, UWC candidates exhibit higher numbers and total amounts of transactions as the typical customer in this segment. To keep the number of returned results manageable for manual analysis, the thresholds used for UWC detection are typically very restrictive. The case study at hand was aimed at evaluating the potential of TMatch in finding interesting interconnections between UWC accounts, that is, highly connected components which may indicate company conglomerates. As a considerably larger amount of

Numer of components	Size
1	982
1	39
1	11
1	10
3	9
1	8
3	7
4	6
5	5
9	4
Number of components 29	Number of nodes within components: 1081 (5%)

Table 4.7: Base UWC components

data can be analyzed in comparison to the manual process, filter settings for UWCs may be defined more broadly, reducing the risk of excluding true UWCs from investigation. Two sets of UWC accounts were used in this case study:

- Base UWC set

  This set corresponds to a relatively broad definition of UWC which was developed at Alphafin and later narrowed down to lower the number of returned results. It contains approximately 22000 UWC candidates.

- Core UWC set

  This set represents the narrowed down definition of UWCs. The core set is a subset of the base UWCs with especially high values in turnover and other activity measures. The core set contains approximately 3800.

Results

In a first run, the base UWC set was used as a base view, while the core set served as the StartSet. With a minimal component size of 4, a total of 29 components was identified (Table 4.7). One of those component exhibits a size of over 900 nodes. It however, exhibited a low score for density, indicating that it represents a number of possibly dense substructures merged by occasional transactions which may not represent a relationship of adequate importance for this analysis. This suggests further experimentation with TMatch configuration, for example by increasing the min-

imal amount or number of transactions for a link to be considered. Purposeful experimentation required a detailed assessment for this component which was infeasible in the course of this case study. The second largest component is of size 39 and clearly outstanding in terms of density. It is illustrated in Fig. 4.15.

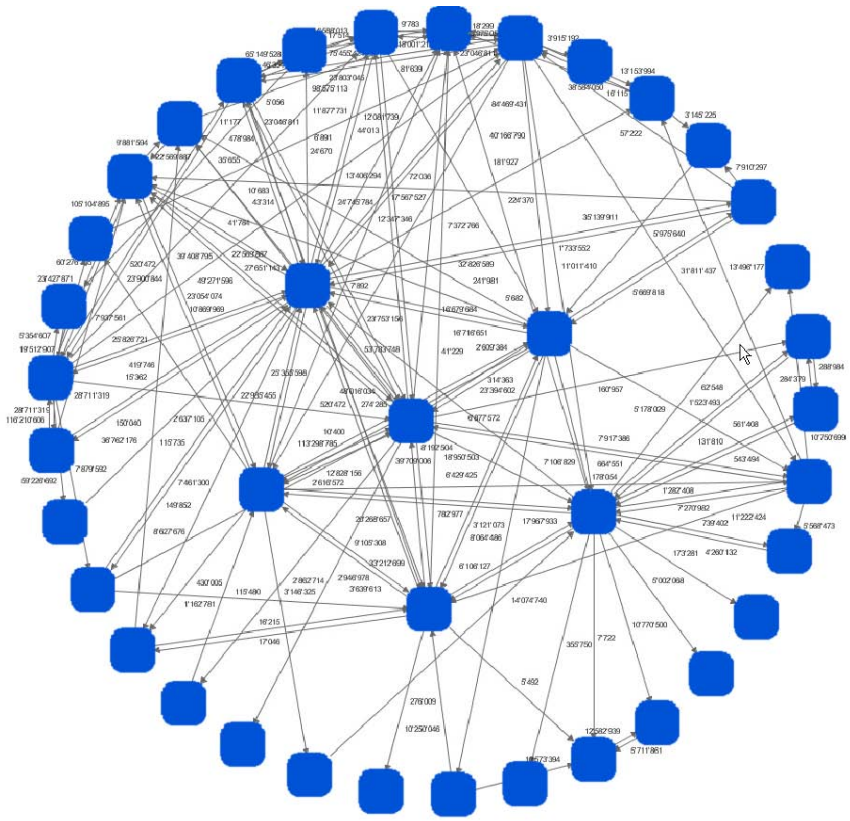


Figure 4.15: A UWC component with very high density

This structure was confirmed by AML experts to be a conglomerate which was previously identified in manual analysis. We were reported that experts had identified approximately 100 accounts belonging to this structure. The fact that TMatch only found 39 of them indicates that our definition of the base UWC set may be too restrictive and motivates further experimentation by adapting the definition of the base graph and the configuration of TMatch.

The second run was solely based on the core UWC set as a base graph without focus on a dedicated start set. Setting the minimal component size to 5 resulted in 40 components. To evaluate the commonness of small components, the numbers of components of size 4 and 3 were addition-

Number of components	Size
1	61
1	41
1	39
1	29
1	27
1	19
2	17
1	15
1	13
1	12
1	11
3	9
3	8
6	7
8	6
8	5
Number of components 40	Number of nodes within components: 48

Table 4.8: Core UWC components

ally calculated.The number of 36 components with size 4 and 70 components with size 3 indicates that smaller components are more common, but still infrequent.

Figure 4.16 shows the distribution of density[7] for the identified components. It reveals that very few components have a very high density. Even among the five most dense components, density decreases substantially from component (a) to component (e) as can be seen in figure 4.16. Sample reviews of some accounts in structure (b) showed conform beneficial owners within the structure which was expected by the responsible expert. A case where more than one beneficial owner was involved in one of the most dense structures was reported to be of particular interest. This led to the considerations of possible applications of TMatch mentioned in 4.4.3. Not the most dense, but biggest components are pictured in figure 4.18. In (a), subgroups of clustered accounts are clearly identifiable. While in (b) and (d), the central accounts are mainly connected via "peripheral" accounts, there exist direct connections between the nodes with high centrality — (e) exhibits a star structure with one prominent node.

[7]Network density denotes the proportion of edges in an network relative to the total number possible $\frac{E}{N(N-1)}$ where E is the number of edges and N is the number of nodes in the graph.

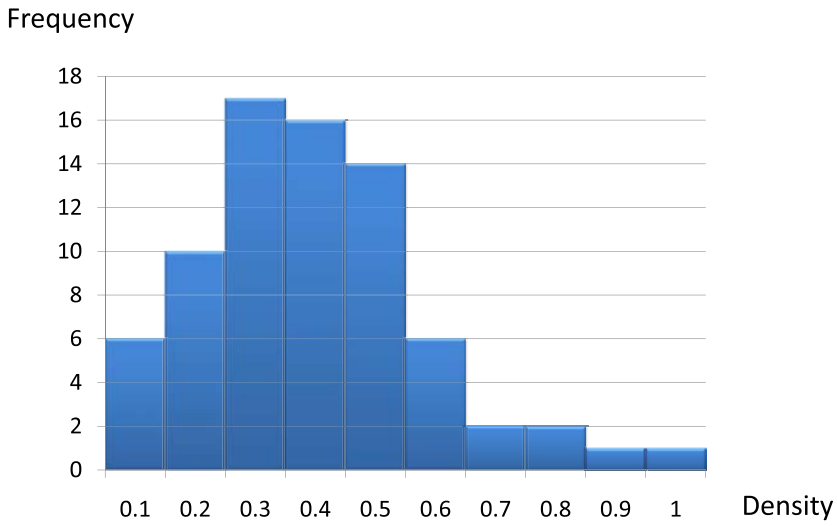


Figure 4.16: The density distribution of core components

4.4.3 Discussion

Number of Results Produced

In all experimental TMatch runs, the number of results produced suggest that the structures looked for occur in the data, but are highly infrequent, which was previously unknown. Of course, the number of results is highly dependent on the exact configuration, but the chosen setting seem to be a reasonable starting point. Figure 4.19 shows the ratio of nodes being part of components $\frac{N_c}{N_a}$ where N_c is the total number of nodes in all identified components and N_a is the total number of analyzed nodes. The numbers suggest a substantially higher connectivity between UWC accounts in comparison to the more heterogeneous population in the region-based analysis. The core UWCs in turn are more likley to be connected among each other than the broader defined base UWCs. This corresponds to the existing business assumptions of the characteristics of UWCs, which is originally based on few sample cases. The fact that N_c is typically substantially smaller than N_a motivates the following considerations when comparing TMatch to currently implemented methods:

As in internal fraud detection, existing detection methods in AML and compliance heavily

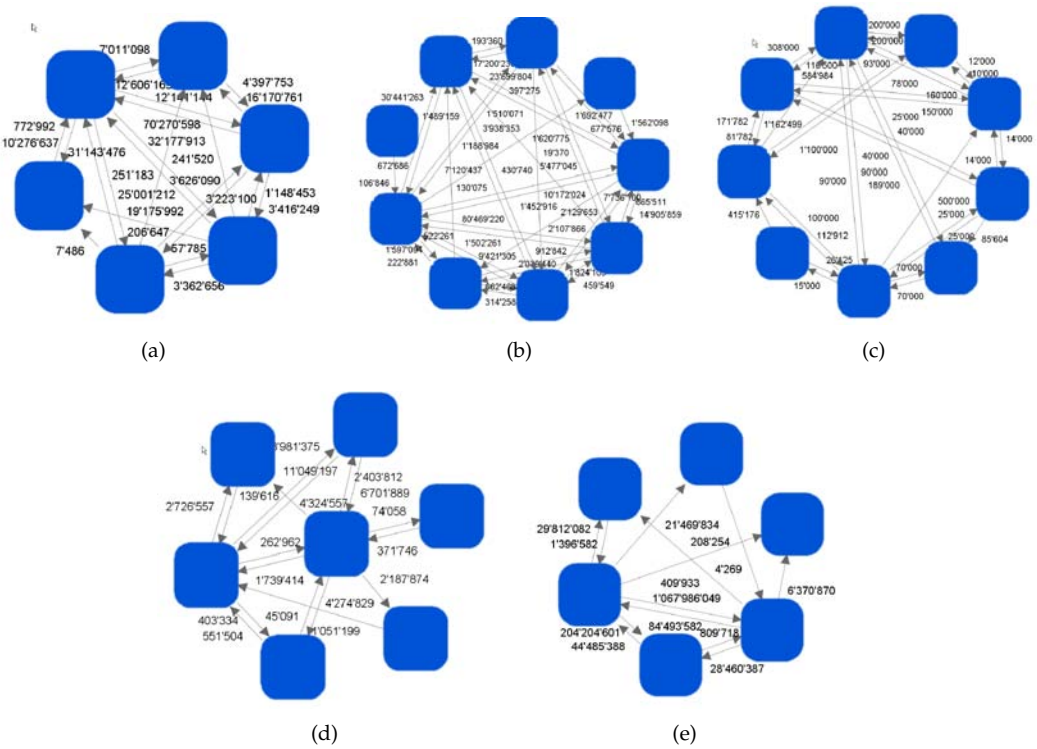


Figure 4.17: The top five density structures

rely on filters focusing on single accounts. To keep the number of alerts at a manageable level, filters are configured very restrictively. Threshold setting is therefore typically motivated by the number of returned results, and not by model knowledge on money laundering patterns. For example, only accounts with a cash turnover of more than 2 million per year may be investigated as a lower threshold may produce too many results. If this can be done without considerably lowering recall remains an open question. As connected components (with dedicated properties concerning minimal size, density and so on) seem to be infrequent, the basic population analyzed may be substantially larger. While a population of 20'000 accounts with a cash turnover of more than 500'000 per year may be out of scope for manual investigation (which often implies analyzing the accounts connections), it can be readily presented to TMatch, which may return a manageable number of interesting structures consisting of accounts in the basic population. These considerations are based on the assumption that the information on relations between accounts of interest generates an added value for investigation. To prove the validity of this assumption is

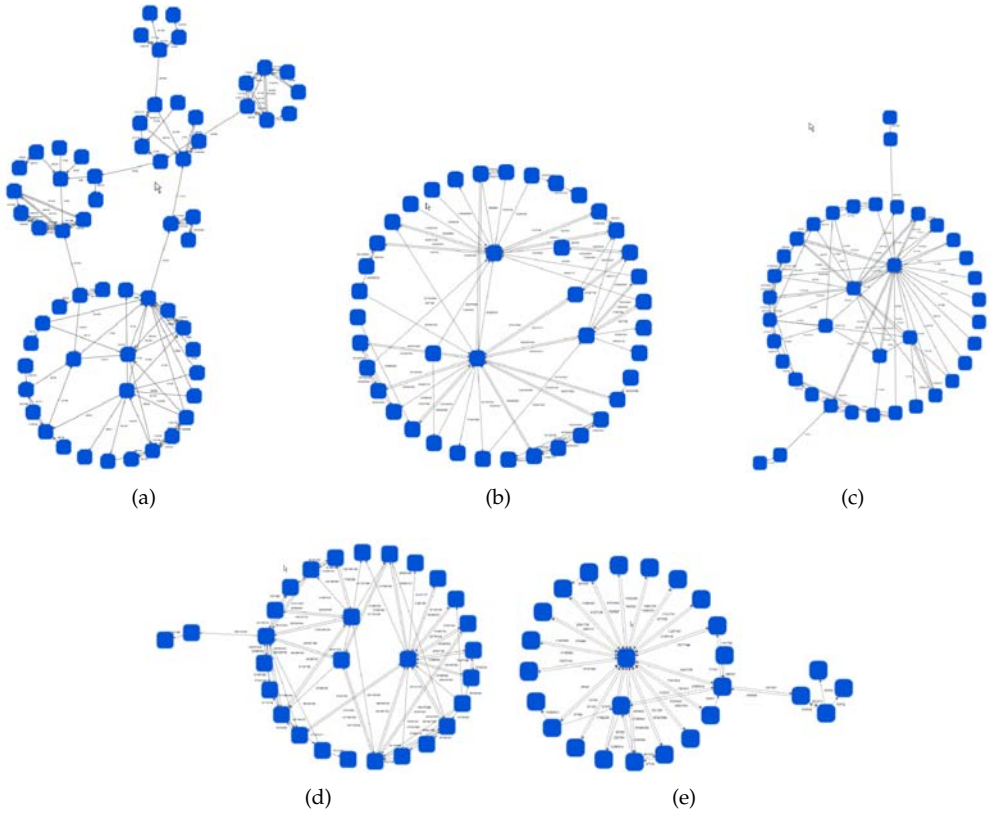


Figure 4.18: The top five size structures

a complex task and may require extensive working experience. Some statements concerning this issues are given in the following section.

Expert Statements

In the explorative case study II, the responsible AML expert looked at a number of selected results from both the high value path and sum chain analysis and reported his first impressions. It was confirmed that the identified structures were of high interest and suggested a more detailed investigation. It was further confirmed that the use of TMatch could lead to insights in terms of occurring structures which is not possible in similar coverage with the current process. Part of the produced results were used for an audit report. In the more specific case study III, we discussed a number of results with the expert responsible for the UWC project. The findings in this discussion are as follows:

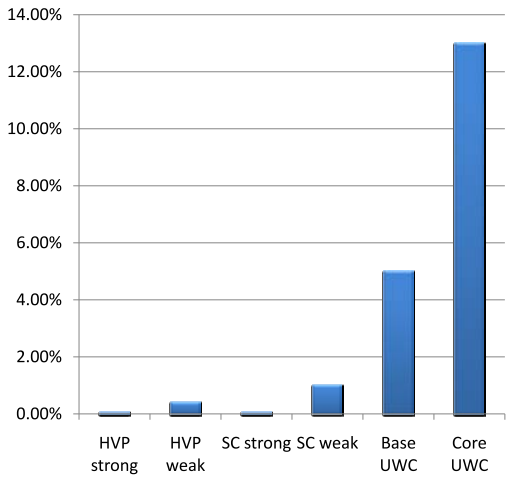


Figure 4.19: The ratio of nodes in components for the different TMatch runs

The focus on patterns of interrelated accounts must not lead to the disregard of accounts with no or only little interaction with other intra-company accounts in investigation. It goes without saying that "internally isolated"accounts may be likewise involved in activities of regulatory interest. Money launderers , for example, may purposely distribute their activities beyond as many company borders as possible to reduce traceability. On the other hand, the following issues motivate automated analysis of intra-company structures:

- Currently established detection methods in Alphafin ignore intra-company structures. The assumption that money laundering or other activities of regulatory interest never happen within a fincancial insititution may be true or false. The results produced in this case study suggest that ignorance of those structures may not be justified.

- Manual identification of the relations between accounts of interest is a crucial element in the manual investigation of alerts from existing detection systems. This step can be partially automated by TMatch. With improved data quality for transactions from and to external parties, unique external accounts can be readily included in a TMatch analysis.

- Regulators may claim that it is particularly critical if a money laundering activity goes undetected where a significant proportion of the underlying structures lies within a company.

Generally speaking, a financial institute is “responsible ” for its customers, which also includes activity among them.

- As seen in the example of UWC analysis, not only explicitly illegal activity may be of interest. Risk/return ratio estimation may exact more detailed understanding of the characteristics of a customer type, which may be supported by TMatch and TVIS.

- Matching automatically identified structures against the information in according customer history entries may reveal if the responsible customer advisor is clued up to the customers activities or if the KYC principle is not satisfied.

We repeatedly mention that a pattern was “confirmed to be of interest for experts”. Admittedly this statement is very imprecise. Given the complexity of the topic and the limited expert resources mentioned before, a more detailed assessment could generally not be done without pretending more information than is actually there.
The case studies are able to indicate potential. The generation of precise quantitative results on the performance of the system requires years of experience in productive use by dedicated analysts as the examples ADS [Kirkland et al., 1998] and FAIS [Senator et al., 1995] show.

Estimated Cost Savings

Current compliance checks typically involve elaborate manual investigation of relationships with other customers and external accounts based on tabular data. We were repeatedly told that currently, the developement of an overview concerning a customers direct and indirect transactional environment may take estimated 3 to 5 days in the presence of moderate complexity. In contrast, TVIS provides an equivalent overview within minutes. An adequate TMatch analysis further automates the investigation process and may improve efficiency. While it has to be stated that enabling the application of TVIS and TMatch in compliance checks will require considerable employee training and even rough cost saving estimation is not possible with our limited knowledge, the potential in raising efficieny is evident. A recent analysis involving 140 accounts was reported to have taken months. After learning about TVIS and TMatch, an expert at Alphafin estimated that the same analysis could have been done within a few days.

4.5 Synthetic data evaluation

4.5.1 Criticism

A number of publications discusses the generation of simulated data for the training and evaluation of fraud detection systems [Lundin et al., 2002; Barse et al., 2003; Cohen and Morrison, 2004]. Where real world data is not available, using simulated data is the only way to evaluate a system. In the case of (supervised) data mining approaches, the simulated data is used to train a system for (assumed) later use with real world data [Barse et al., 2003].
However, if the knowledge is available for simulating fraud cases in sufficient quality, it may be more effective to convert it to model knowledge and build a pattern-matching based detection system instead of making the detour of calculating a model.
Apart from that, generating high quality simulation data may be a very demanding task. Based on the proposals in [Lundin et al., 2002], we designed a basic transaction simulator for the reproduction of real world payment networks based on a seed of analyzed real world data.
The Java Transaction Simulator (JTS) works as follows: A subset of real world data is chosen and serves as a seed (as proposed in [Barse et al., 2003]).

In a first step, we summarized the transactions between each pair of accounts in the seed. Consider the transactions shown in the upper left of Figure 4.20 as an example. There are 3 transactions from account A to account B and 1 transaction from Account B to A. The behavior between those two accounts could, hence, be summarized as $\{3, -1\}$. To additionally include information about transaction amounts, the transactions were bagged. Assuming two bags (bag_1 $\in$ 1-500\$, $bag_2 \in$ 500-1000\$) a summary of $\{2bag_1, 1bag_2, -1bag_1\}$ refines the previous summary with transaction size information.

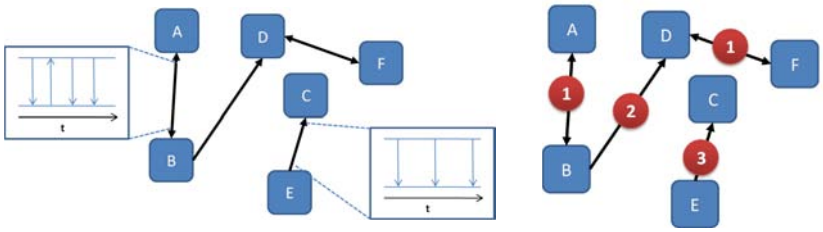
Second, we clustered the pairwise behavior patterns according to their summaries. Each cluster (shown as red numbers in Figure 4.20 on page 134) represents a generalized behavior between two accounts, e.g. ”numerous small transactions” (type 1) or ”few big transactions” (type 2).

Third, we replaced all the transactions between two accounts with the corresponding pairwise behavior cluster number (shown in Figure 4.20 on the upper right). Hence, we characterized each account by the bag of behavior types it is involved in.

Fourth, we clustered the accounts according to their behavior characterization resulting in

groups of account types (green in Figure 4.20).

To generate the synthetic data set, we would now generate the desired number of accounts with the same distribution of account types as given by the seed (i.e., the distribution of accounts in the seed per cluster denoted by the green number). For each account, relationships are generated using the distribution of behavior observed in its cluster (i.e., according to the distribution of red numbers in the account's cluster). The result is a transaction network with a behavioral distribution akin to the one of the seed.



Account	„In“	„Out“	
A		1x 1	1
B	1x 1	1x 2	2
C	1x 3		3
D	1x 2 ,1x 1		4
E		1x 3	5
F		1x 1	1

Figure 4.20: Synthetic data generation with JTS

Implementation and a simple evaluation is described in [Galliker, 2008]. Experiments with real world data seeds showed reasonable results, but also made clear that extensive further development and improvements are necessary for the simulation of realistic payment behavior of a highly heterogeneous and complex population. Due to the issues mentioned above and the fact that real world data was available for evaluation, we decided to abandon work on simulated data improvement and to limit synthetic data evaluation on runtime.

Publications introducing new pattern matching algorithms heavily rely on runtime evalua-

tions [Gallagher, 2006b]. In our case, the focus does not lie on the theoretical aspects of the pattern matching algorithms, but on their application in an information system solving a real world problem. Therefore, our approaches are not optimized in terms of runtime performance, but in terms of feasibility under given real world conditions. The achievement of this objective cannot be measured by means of algorithm runtime. Additionally, the actual runtime in our case studies in Alphafin is primarily determined by factors beyond the system and the area of our influence. In particular, data base and network speed are crucial factors. Highly varying load, changing data base index structures and connection interruptions make accurate runtime evaluations impossible. The following runtime evaluations are therefore of limited expressive power and are rather given for the sake of completeness.

4.5.2 ChainFinder Runtime Evaluation

While the Java-based implementation of the ChainFinder with its incremental loading of data using lightweight SQL-statements and intermediary results logging was the preferred choice when analyzing real data at Alphafin, the SQL based implementation features a considerably better runtime performance when evaluating synthetic data. This is evident as the extensive communication with the DBMS in incremental loading leads to a considerable overhead. If the limited functionality and configurability of the SQL version is sufficient and the infrastructure is resilient and stable enough to reliably handle complex queries for the amount of data analyzed, runtime can be reduced considerably. Figure 4.21 shows the average runtime of the two ChainFinder versions for approximately 370'000 transactions generated with JTS for different numbers of analyzed chain root nodes. The difference in runtime is notable.

4.5.3 TMatch Runtime Evaluation

TMatch was evaluated on different sets of random transactions between varying numbers of accounts. Figure 4.22 shows overall time consumption of TMatch for different numbers of edges. As the x-coordinate shows the node-to-edge ratio, the respective graphs get more sparse when moving from left to right on a plot line. However, as the graph is a multigraph, network density can only be approximated. The figure shows that in particular for large numbers of edges, runtime

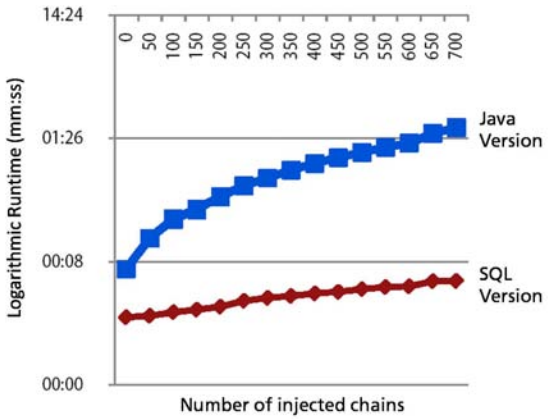


Figure 4.21: Chainfinder runtime evaluation

is low for a very sparse and an approximately fully connected graph as the number of identified components is very low in both cases. There is a clear peak at a node-to-edge ratio of 0.1, e.g. 100'000 nodes and 1 million edges in a multigraph. In a real world scenario, the base graph and search options will typically be chosen such that components are found but occur relatively rarely, so the two extreme cases will never be met in practice.

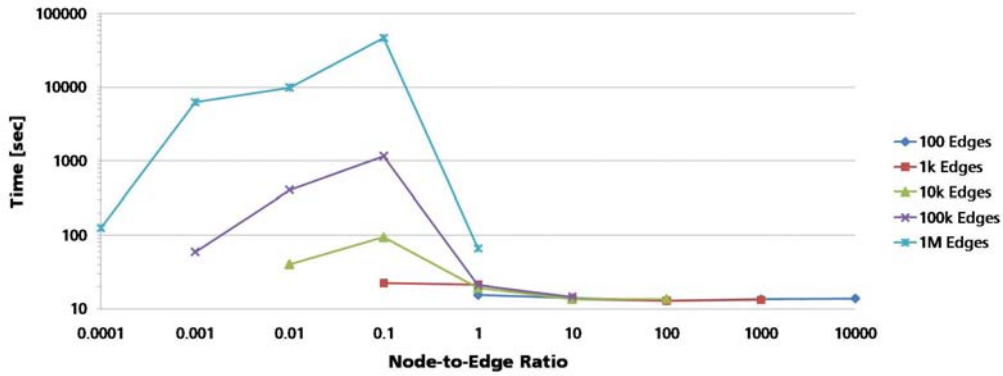


Figure 4.22: The TMatch node-to-edge ratio time consumption graph

Figure 4.23 shows time and memory consumption for the typically most costly expander component in its three implementations RAM, DB and ThreadedDB. While memory consumption is unsurprisingly considerably higher for the RAM based version, it is interesting to see that the ThreadedDB implementation exhibits a better runtime performance where the graph tends to be relatively sparse.

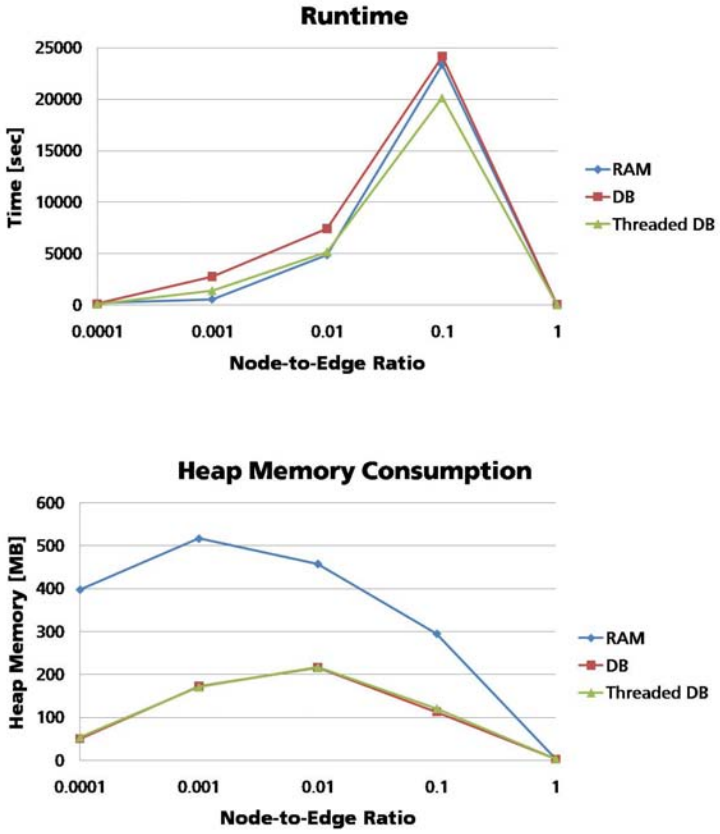


Figure 4.23: The expander node-to-edge ratio to time consumption graph with 1M edges

5

Future Work

Possible and promising ways to go from where we are today are manifold. In the following, selected issues are discussed.

5.1 Extended TVIS User Group Customization

The requirements for the visualization component vary between different potential user groups which were identified in the course of the project. In internal fraud, fine-grained analysis of relatively few data at a time is common. AML-related investigations typically require more data, but an aggregated transaction view may be sufficient. An expert using TVIS as an investigation tool needs more elaborate features than a client advisor requesting visualizations for KYC. Given a number of possible scenarios for the productive use of both TMatch and TVIS, customization and further integration into business processes becomes necessary. Supporting the transformation from research stage to a productive solution, which partly already has taken place — in particular for the visualization component — is part of future work.

5.2 Automated Exclusivity Scoring

As described in the internal fraud case study, we found that structures almost solely resulting from one customer advisor using MTT transactions below the defined operational threshold may be a valuable indicator for possible fraud. The manual process of structure exclusivity assessment

could be automated by an appropriate scoring model. TMatch may be readily extended with exclusivity scoring. This has not been done due to the focus switch mentioned above in the course of the project. Of course, exclusivity scoring may not only be used in combination with Chain and Smurfing structures, but with a wider range of structures in terms of transaction types, involved accounts and so on.

5.3 Structural Peer Group Analysis

Peer Group Analysis detects changes in behavior in relation to a set of similar instances, the peer group [Weston et al., 2008]. This is typically done on the single account/ aggregated events (in our case transactions) level. Peer Group Analysis could be combined with relational pattern matching to profile and monitor customer advisors with regard to the use of transaction patterns of interest. For instance, the average number of well-grounded, legal transaction chains may vary depending on the set of customers an employee is in charge of. Client advisors with similar customer sets in terms of those transaction patterns could build a peer group to account for that fact. According to the nature of the data, peer groups may be defined by organizational structures (if they correspond to similar pattern frequency) or calculated in a prior clustering. This approach could also allow for a straightforward integration of existing monitoring results as additional features for the peer group analysis. For illustration, we propose a possible minimal feature set for each client advisor in Peer Group Analysis:

- number of Single/Sum Transaction Chains (ChainFinder scores)
- structural relationships between occurring Single/Sum Transaction Chains (TMatch scores)
- structural relationships between all triggered transactions of type *MTT* (TMatch scores)
- exclusivity scores (see above)
- results of existing monitoring results (details are confidential)

As in traditional peer group analysis, significant deviations from the behavior of the peer group could be detected and trigger an alert. It has to be noted that the definition of significant deviation

may not be a trivial task, in particular for the structural features. As mentioned above, we repeatedly experienced that the transaction network at hand is highly heterogeneous. Abrupt behavior changes of customers (and therewith of their customer advisors) seem to be quite common — a characteristic which handicaps the use of Peer Group Analysis as argued above. Extensive analysis is necessary to decide if the inclusion of transaction patterns which suggest a high discriminative power (as transaction chains) may resolve this problem. We implemented an experimental version of this approach. However, it soon became clear that because of extensive reorganizations and renaming afflicting the relevant data, a reasonable evaluation was not straightforward. The necessary extensive support from Alphafin was not available, in particular because the demanded research focus had switched to other topics in the meantime. Therefore, an appropriate implementation and evaluation of this approach remains future work.

5.4 Connection Probability

In both fraud detection and AML, unusual connections between accounts are of special interest. In the existing fraud investigation process, experts make use of their experience to spot connections which seem unusual in visual analysis. The customer history information of involved accounts is then checked for hints explaining the transaction(s) in question. For instance, recurring high amount payments from a machine manufacturer to a component supplier may be perfectly plausible, while the same payments may require a closer look if the beneficiary is, say, a tourist agency. A possible way to support this procedure is the automated identification of unusual connections[1] , in terms of their mere existence or in terms of their characteristics. This approach has been considered but was prohibited by the lack of high quality industry type codes or similar information available. There are efforts to integrate according data fields into the data warehouse. Identifying outliers in terms of connection probability may be a valuable complement to other fraud detection measures. A first implementation may focus on a single-edge representation of the transaction graph for probability calculation and therefore be limited to aggregated transaction amounts. However we expect that more detailed information should be used for reasonable results — in particular height and volatility of payment amounts, number and regularity of single

[1] This problem is related to the field of link analysis

transactions and similar.

6

Limitations and Conclusions

6.1 Limitations

In [Whitrow et al., 2009], the authors mention that, unavoidably, "any particular [fraud] study is always a snapshot in time and space" and many of their conclusions "will therefore be tentative". This is undoubtedly also true for our work. The collaboration with an industry partner allowed us to gain numerous practical insights and approach the subject from a pragmatic view. However, it entails that the proposed solutions are tailored to our industry partner and generalizability may be limited.

Heavily based on available model knowledge, the presented approach is, by definition, not ideal for finding completely new (or unknown) fraudulent behaviour, although the work with TVIS showed some potential for exploring previously unknown patterns.

We focused on Chain Transactions and Smurfing in the design of our approach[1]. Of course, we cannot rule out that fraud is possible without generating any of those patterns (while avoiding existing threshold monitors as well). If fraudulent transactions are indistinguishable from normal transactions in respect of all the information in the available data, all data-based fraud-finding solutions fail.

[1] It, however, has to be stated that the ChainFinder — in spite of its name — may be adapted to match completely different patterns where the relations between attribute values are more important than specific values.

6.2 Conclusions

In this thesis, we consider a wide variety of analytical fraud detection approaches and discuss their applicability in different settings by proposing a unifying framework.

We introduce the special case of internal fraud and its characteristics in the collaborating financial institution. A system based on graph pattern matching and visualization is proposed. We evaluated the potential of the approach in example case studies at Alphafin.

The visualization component was transferred from an experimental prototype to a productive tool in the course of this thesis. Two releases were rolled out at Alphafin, a further releases is planned for the following year, and the productive integration of the detection component is currently discussed.

Anyhow, we were not able to proceed as fast and far as desired. The circumstances were challenging for conducting research. Substantial work was required to set the stage such as designing and implementing the visualization component, design and optimization of table views and analyzing undocumented data properties, to just name a few. During the first two years of the project, we did not get direct access to the data, but had to rely solely on expert input. We were bound to the available infrastructure and resources in Alphafin, which limited our action flexibility. Personal fluctuations and the fact that no business expert resources were officially allocated to support the one-man project was severe in particular in the lack of labeled examples (identification knowledge). As a consequence, expressivity of evaluation was limited.

At the same time, these aggravating circumstances were appealing for a thesis on the interface between research and industry.

Both the design and the implementation of our approaches were focused on the pragmatic challenges we met. This led to a relatively straightforward solution.

As stated earlier, we do not see our main contribution in developing yet another sophisticated detection approach, but in considering the different perspectives and goals in research and industry — and finding a way to combine the two.

A

Implementation

A.1 TVIS Implementation

TVIS consists of a java server and a client. The server is running on Apache Tomcat and is responsible for the communication with the database and processing of the data for the clients. The client is based on the MVC-pattern and makes use of a third party visualization library (YFiles in a first version, Tom Sawyer in later version for organizational reasons). A prototype was delivered to Alphafin, further developments were done at Alphafin and resulted in a closed source software.

A.2 ChainFinder Implementation

The ChainFinder mainly used in this thesis is implemented in Java, making use of JDBC connections. A faster, but less flexible implementation was done in SQL. Both implementations are described in a nutshell in the following sections.

A.2.1 Java Implementation

In the following, a short description for the most important classes of th e Java implementation are given.

Field	Example Value	Description
String CHAIN_TYPE	"SINGLE"/"SUM"	Type of chains to be matched
String WINDOW_SIZE	"7"	In days
String MINIMUM_CHAIN_LENGTH	"2"	In number of transactions
String DIRECTION	"FORWARD"/ "REVERSE"/ "BIDIRECTIONAL"	Search direction; root nodes are either sources, targets or intermediaries in chains
boolean RANDOM_NODE_ID	true/false	Random structure identification (RSI)
boolean MAX_CLONE_NUMBER	4	Max of allowed clones for RSI
double FUZZY_EQUALS_MIN_RATIO	0.9	For fuzzy amount matching
double FUZZY_EQUALS_MAX_RATIO	1.1	For fuzzy amount matching
boolean REGISTRATOR_TRACK	true/false	Check for same registrator within chains

Table A.1: ChainFinder configuration fields

algorithm.ChainFinder

A singleton class controlling algorithm runs via a start method that takes the root node as an argument and ensures that successor nodes are expanded as long as there are any.

data.SingleChainNode, data.SumChainNode

These two classes implement the data.Node interface and represent a node in single and sum chain mode respectively. For example, a difference between the two types of nodes is that SumChainNodes provide a method to calculate the TimeLineScore.

data.SingleChainParticle, data.SumChainParticle

These two classes implement the data.Particle interface and account for the different attributes and behaviours of a particle, in particular in updating, cloning, and logging information.

data.Configuration

A simple configuration class for setting ChainFinder behaviour. The configuration information in this class may be readily managed by property files or a graphical user interface. The following table gives a short overview of the most relevant configuration fields in ChainFinder. Additional configuration fields for specifying the DB connection, tables and field names as well as prepared SQL statements are given but not listed here.

Utility classes

A number of classes is responsible for managing DB connections and looging functionality (tools.DBConnector, tools.LogDBConnector,data.ResultLogger)

A.2.2 SQL Implementation

A simple, fast implementation of the ChainFinder can be done in SQL. Functionalty limitations may be resolved by using more expressive languages as PL/SQL (which was prohibited in our setting). A basic SQL query for finding a chain of length 2 may look like this:

*Select * from root.trx as trx1, root.trx as trx2*
where trx1.ORIG = 'A' and
trx1.BEN = trx2.ORIG and
*trx2.AMOUNT between (trx1.AMOUNT*0.9) and (trx1.AMOUNT*1.0) and*
trx2.DATE between (trx1.DATE) and (trx1.DATE + 5 days);

A.3 TMatch Implementation

TMatch is also implemented in Java, making use of Ibatis for the mapping of SQL DBs and objects in Java. Main packages are the following:

NodeSetExpanders

A number of NodeSetExpanders is avaliable for expanding nodes in the search of interesting structures. Threaded and non-threaded implementations are available for DB-based and RAM-based expansion.

ComponentFinder

This class is responsible for the identification of components in a graph

AlertGraph

This class represents the AlertGraph holding pattern matching results from the pattern matching steps

ScoreModels and Scorers

Classes in this family manage the scoring of components. New score models can be added dynamically. A description how this is done is provided in [Moll, 2009]

Utiliy classes

A number of utility classes are available for Unit Test, Logging, Data organization and other required support functions.

Configuration

The configuration of TMatch is organized in a number of property files. Default properties are given in TMatch.default.properties

A.4 TMatch Graphical User Interface

A simple Graphical User Interface has been implemented for demonstration reasons. After presented a welcome screen outlining the overall process (Figure A.1), the user is guided through the basic configuration of the search. A number of predefined search profiles can be chosen (Figure A.2). In expert mode all settings can be configured directly. (Figure A.3) Based on expert configuration, the creation of new search profiles for the normal user is possible. Results, that is, retrieved components are shown in the result display. The total score field can be expanded to reveal the detailed scores (Figure A.4). A simple visualization function is provided - detailed investigation is done in TVIS.

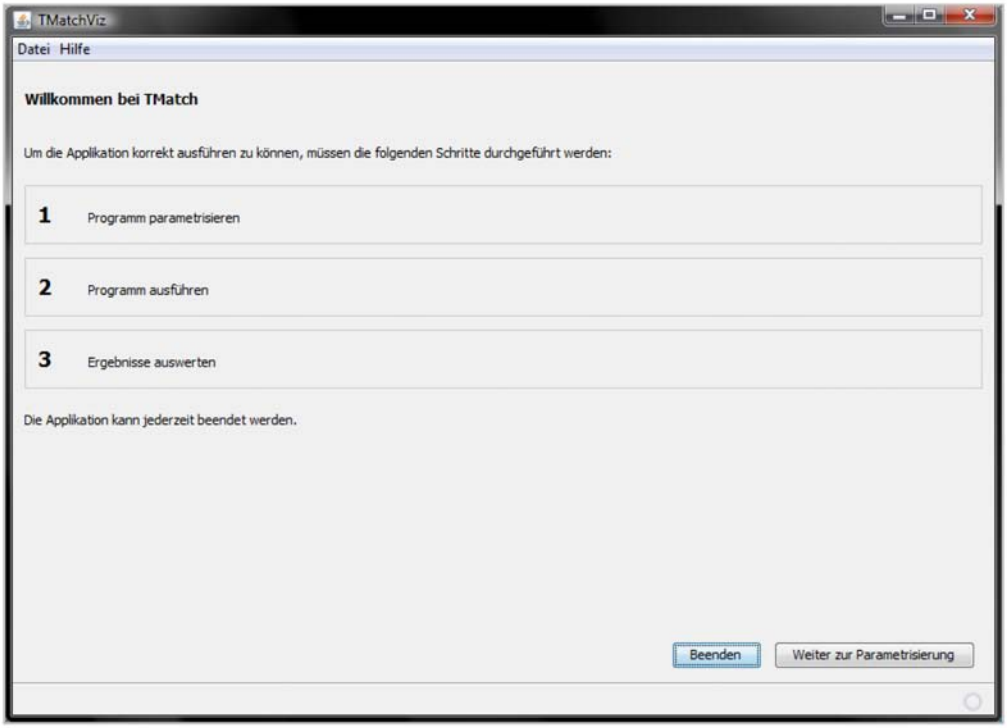


Figure A.1: TMatch GUI welcome screen

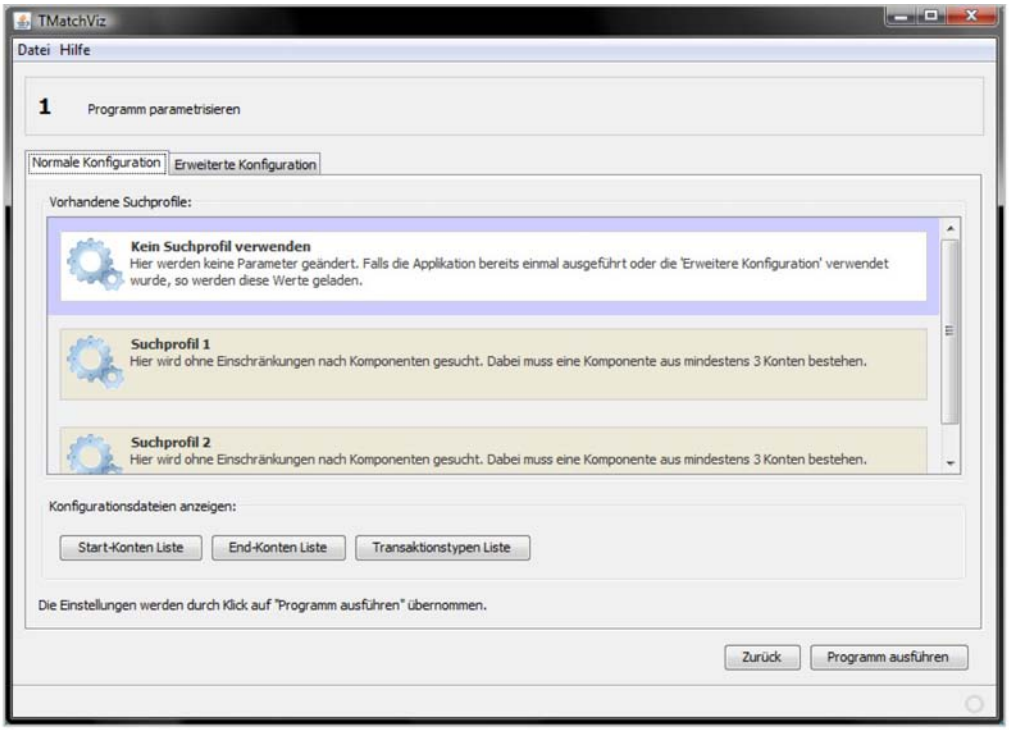


Figure A.2: TMatch GUI search profile configuration screen

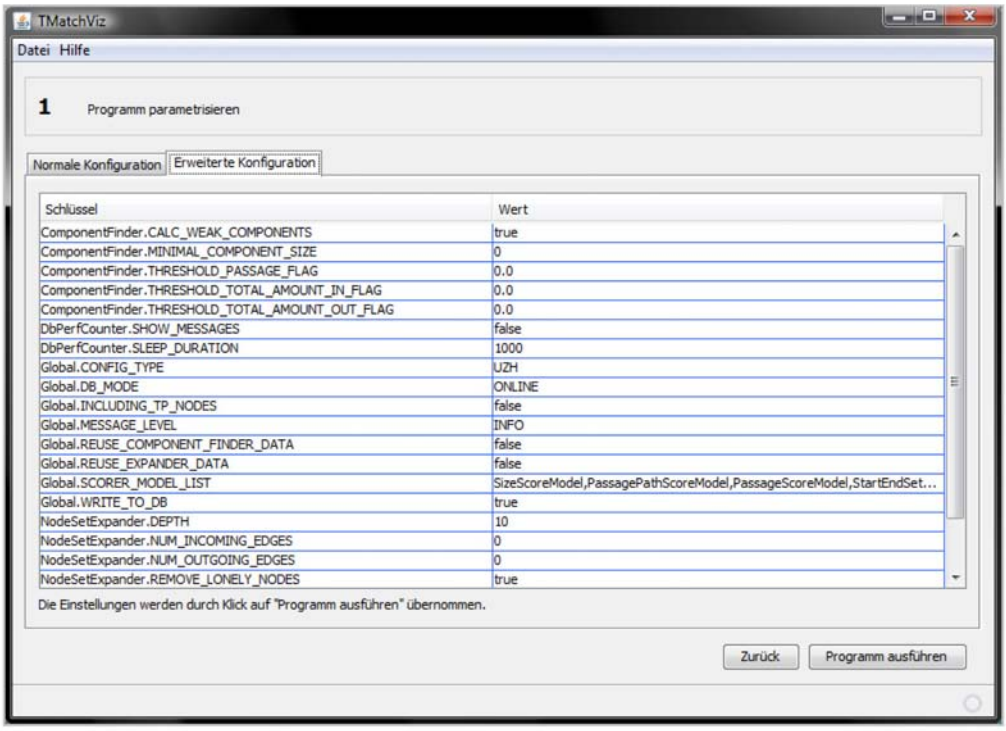


Figure A.3: TMatch GUI expert configuration screen

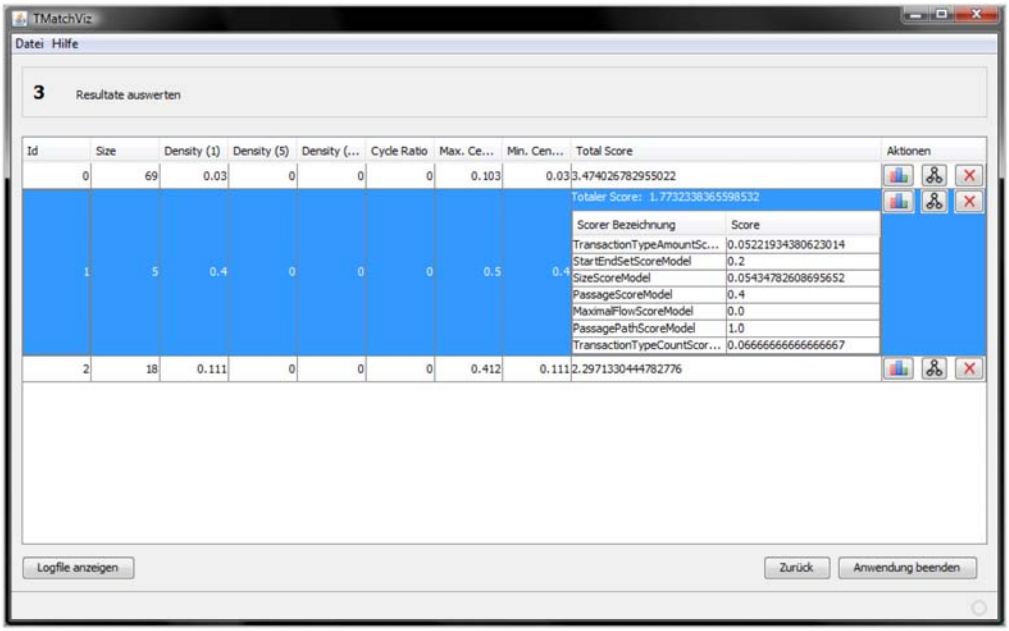


Figure A.4: TMatch GUI result display screen

A.5 Example TMatch Score Results

Table A.2 and A.3 give detailed TMatch results as discussed in case study III (Section 4.4.2 on page 124). Due to the high level explorative character of first experiments and performance reasons, the calculated scores were limited to a minimal subset of subgraph[1] size, degree density and maximal degree centrality. In table A.3, displayed components are limited to a size > 5.

[1] denoted "component" in the scoring

Component	Size	Degree Density	Max Centrality
0	982	0.001	0.042
1	10	0.144	0.333
2	39	0.101	0.408
3	9	0.139	0.25
4	3	0.333	0.5
5	5	0.3	0.625
6	6	0.233	0.3
7	8	0.25	0.643
8	9	0.25	0.5
9	5	0.25	0.5
10	3	0.333	0.5
11	5	0.35	0.625
12	5	0.3	0.5
13	11	0.127	0.5
14	3	0.667	0.75
15	7	0.238	0.583
16	3	0.833	1
17	5	0.2	0.375
18	6	0.367	0.6
19	4	0.333	0.5
20	3	0.333	0.5
21	3	0.5	0.5
22	4	0.5	1
23	3	0.333	0.5
24	3	0.5	0.75
25	3	0.833	1
26	7	0.381	0.583
27	3	0.333	0.5
28	4	0.5	0.833
29	4	0.25	0.5
30	9	0.111	0.25
31	3	0.833	1
32	3	0.667	0.75
33	6	0.167	0.5
34	4	1	1
35	3	0.667	0.75
36	3	0.333	0.5
37	3	0.667	0.75
38	3	0.333	0.5
39	6	0.267	0.4
40	4	0.333	0.667
41	3	0.667	0.75
42	3	0.333	0.5
43	3	0.5	0.75
44	4	0.5	0.667
45	4	0.417	0.5
46	3	0.333	0.5
47	3	0.333	0.5
48	3	0.333	0.5
49	7	0.143	0.333
50	4	0.333	0.667

Table A.2: All UWC component scores

Component	Size	Degree Density	Max Centrality
0	12	0.121	0.409
1	7	0.214	0.417
2	13	0.096	0.208
3	17	0.121	0.594
5	29	0.039	0.188
6	39	0.056	0.342
10	27	0.088	0.731
11	15	0.105	0.286
12	8	0.268	0.857
17	61	0.03	0.083
18	9	0.542	0.875
20	41	0.059	0.388
23	9	0.333	0.563
24	7	0.214	0.333
27	11	0.364	0.55
30	9	0.472	0.688
31	17	0.154	0.406
36	19	0.137	0.5
40	7	0.238	0.5
43	8	0.179	0.357
48	6	0.333	0.6
49	6	0.2	0.5
59	8	0.143	0.286
60	6	0.267	0.5
64	6	0.333	0.5
70	7	0.238	0.583
77	6	0.733	1
81	7	0.405	0.833
82	6	0.4	0.8
85	6	0.267	0.4
86	6	0.2	0.4
141	7	0.238	0.417

Table A.3: Core UWC component scores (Size $>$ 5)

List of Figures

List of Tables

Bibliography

[Aleman-Meza et al., 2005] Aleman-Meza, B., Halaschek-Wiener, C., Sahoo, S. S., Sheth, A. P., and Arpinar, I. B. (2005). Template based semantic similarity for security applications. In Kantor, P. B., Muresan, G., Roberts, F., Zeng, D. D., Wang, F.-Y., Chen, H., and Merkle, R. C., editors, *ISI*, volume 3495 of *Lecture Notes in Computer Science*, pages 621–622. Springer.

[Altenkirch, 2006] Altenkirch, L. (2006). *Techniken der Geldwaesche und ihre Bekaempfung*. Bankakademie-Verlag GmbH, Frankfurt am Main.

[Association of Certified Fraud Examiners, 2008] Association of Certified Fraud Examiners (2008). 2008 report to the nation on occupational fraud and abuse. Copyright 2008 by the Association of Certified Fraud Examiners, Inc., available at http://www.acfe.com/resources/publications.asp?copy=rttn, accessed on January 5th, 2010.

[Bankersonline, 1992] Bankersonline (1992). Internal fraud. Bankers' Hotline, Vol. 3, No. 1, 5/92. Available at http://www.bankersonline.com/articles/bhv03n01/bhv03n01a2.html, accessed on January 5th, 2010.

[Barse et al., 2003] Barse, E. L., Kvarnström, H., and Jonsson, E. (2003). Synthesizing test data for fraud detection systems. In *ACSAC*, pages 384–395. IEEE Computer Society.

[Basel Commitee on Banking Supervision, 2006] Basel Commitee on Banking Supervision (2006). International convergence of capital measurement and capital standards. Avaliable at http://www.bis.org/publ/bcbs128.pdf, accessed on January 5th, 2010.

[Becker, 1968] Becker, G. S. (1968). Crime and punishment: An economic approach. *The Journal of Political Economy*, 76(2):169–217.

[Berendt, 2005] Berendt, B. (2005). The semantics of frequent subgraphs: Mining and navigation pattern analysis. In Bauer, M., Brandherm, B., Fürnkranz, J., Grieser, G., Hotho, A., Jedlitschka, A., and Kröner, A., editors, *LWA*, pages 91–102. DFKI.

[Bernstein et al., 2002] Bernstein, A., Clearwater, S., Hill, S., Perlich, C., and Provost, F. (2002). Discovering knowledge from relational data extracted from business news. In *Proceedings of the KDD-2002 Workshop on Multi-Relational Data Mining (MRDM-2002)*, pages 7–20.

[Bernstein et al., 2003] Bernstein, A., Clearwater, S., and Provost., F. (2003). The relational vector-space model and industry classification. In *IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*.

[Berry et al., 2004] Berry, P. M., Harrison, I., Lowrance, J. D., Rodriguez, A. C., Ruspini, E. H., and Jerome M. Thomere, M. J. W. (2004). Link analysis workbench. Technical report, SRI International.

[Bolton and Hand, 2002] Bolton, R. and Hand, D. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255.

[Brockett et al., 2002] Brockett, P. L., Derrig, R. A., Golden, L. L., Levine, A., and Alpert, M. (2002). Fraud classification using principal component analysis of ridits. *The Journal of Risk and Insurance*, 69:341–371.

[Bross, 1958] Bross, I. D. (1958). How to use ridit analysis. *Biometrics*, (14):18–38.

[Burge and Shawe-Taylor, 2001] Burge, P. and Shawe-Taylor, J. (2001). An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection. *J. Parallel Distrib. Comput.*, 61(7):915–925.

[Cahill et al., 2002] Cahill, M., Chen, F., Lambert, D., Pinheiro, J., and Sun, D. (2002). Detecting fraud in the real world. In *Handbook of Massive Datasets*, pages 911–930. Kluwer Academic Publishers.

[Chakrabarti and Faloutsos, 2006] Chakrabarti, D. and Faloutsos, C. (2006). Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2.

[Chakrabarti et al., 1998] Chakrabarti, S., Dom, B., and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA. ACM.

[Chan et al., 1999] Chan, P., Fan, W., Prodromidis, A., and Stolfo, S. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14:67–74.

[Chang et al., 2008] Chang, R., Lee, A., Ghoniem, M., Kosara, R., Ribarsky, W., Yang, J., Suma, E., Ziemkiewicz, C., Kern, D., and Sudjianto, A. (2008). Scalable and interactive visual analysis of financial wire transactions for fraud detection. *Information Visualization*, 7(1):63–76.

[Clark and Jolly, 2008] Clark, N. and Jolly, D. (2008). Societe generale loses $ 7 billion in trading fraud. Available at http://www.nytimes.com/2008/01/24/business/worldbusiness/24iht-socgen.5.9486501.html, accessed on January 5th, 2010.

[Coffman et al., 2004] Coffman, T., Greenblatt, S., and Marcus, S. (2004). Graph-based technologies for intelligence analysis. *Commun. ACM*, 47(3):45–47.

[Cohen and Morrison, 2004] Cohen, P. R. and Morrison, C. T. (2004). The hats simulator. In *Winter Simulation Conference*, pages 849–856.

[Cook and Holder, 2000] Cook, D. J. and Holder, L. B. (2000). Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41.

[Cortes et al., 2001] Cortes, C., Pregibon, D., and Volinsky, C. (2001). Communities of interest. *Lecture Notes in Computer Science*, 2189:105–??

[Eberle and Holder, 2009] Eberle, W. and Holder, L. (2009). Mining for insider threats in business transactions and processes. In *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, pages 163–170.

[Eberle and Holder, 2007] Eberle, W. and Holder, L. B. (2007). Mining for structural anomalies in graph-based data. In *DMIN*, pages 376–389.

[Edmonds and Karp, 1972] Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264.

[Encyclopedia Britannica, 2006] Encyclopedia Britannica (2006). Encyclopedia britannica online. Available at http://www.britannica.com/EBchecked/topic/217591/fraud, accessed on January 5th, 2010.

[Ezawa and Norton, 1995] Ezawa, K. J. and Norton, S. W. (1995). Constructing bayesian networks to predict uncollectible telecommunications accounts. *IEEE Intelligent Systems*, 11(5):45–51.

[Fanning and Cogger, 1998] Fanning, K. and Cogger, K. O. (1998). Neural network detection of management fraud using published financial data. In *International Journal of Intelligent Systems in Accounting, Finance & Management*, pages 21–24.

[FATF, 2009] FATF (2009). Financial action task force annual report. Technical report, Financial Action Task Force. Available from http://www.fatf-gafi.org. Accessed on 26th July 2009.

[Fawcett et al., 1997] Fawcett, T., Foster, and Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1:291–316.

[Fawcett and Provost, 1996] Fawcett, T. and Provost, F. (1996). Combining data mining and machine learning for effective user profiling. pages 8–13. AAAI Press.

[Fawcett and Provost, 1999] Fawcett, T. and Provost, F. (1999). Activity monitoring: noticing interesting changes in behavior. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 53–62. ACM.

[Fayyad et al., 1996] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34.

[Fitch, 2006] Fitch, T. P. (2006). Dictionary of banking terms. Available at http://www.answers.com/library/Banking+Dictionary-cid-31665.html, accessed on January 5th, 2010.

[Gallagher, 2006a] Gallagher, B. (2006a). Matching structure and semantics: A survey on graph-based pattern matching. In *In AAAI FS '06: Papers from the 2006 AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection*, pages 45–53.

[Gallagher, 2006b] Gallagher, B. (2006b). The state of the art in graph-based pattern matching. Technical report, Lawrence Livermore National Laboratory.

[Galliker, 2008] Galliker, S. (2008). Generierung von synthetischen Banktransaktionsdaten. Bachelorarbeit im Fach Informatik, Universitaet Zuerich.

[Geiger, 2008] Geiger, H. (2008). Banken und Vertrauen. Farewell lecture.

[Getoor and Diehl, 2005] Getoor, L. and Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12.

[Giugno and Shasha, 2002] Giugno, R. and Shasha, D. (2002). Graphgrep: A fast and universal method for querying graphs.

[Goldberg and Senator, 1995] Goldberg, H. G. and Senator, T. E. (1995). Restructuring databases for knowledge discovery by consolidation and link formation. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 136–141. AAAI Press.

[Greenblatt et al., 2005] Greenblatt, S., Marcus, S., and Darr, T. (2005). Tmods - integrated fusion dashboard - applying fusion of fusion systems to counter-terrorism. In *Proc. International Conference on Intelligence Analysis*.

[Holder et al., 1994] Holder, L. B., Cook, D. J., and Djoko, S. (1994). Substructure discovery in the subdue system. In *In Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, pages 169–180.

[Hollmén and Tresp, 1998] Hollmén, J. and Tresp, V. (1998). Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In Kearns, M. J., Solla, S. A., and Cohn, D. A., editors, *NIPS*, pages 889–895. The MIT Press.

[Huang et al., 2009] Huang, M. L., Liang, J., and Nguyen, Q. V. (2009). A Visualization Approach for Frauds Detection in Financial Market. In *Information Visualisation, 2009 13th International Conference*, pages 197–202.

[Jeffery Lehman, 2002] Jeffery Lehman, S. P. (2002). Management antifraud programs and controls. Excerpted from Statement on Auditing Standards No. 99, Considerations of Fraud in a Financial Statement Audit. Copyright 2002 by American Institute of Certified Public Accountants, Inc. New York, NT., Available at http://www.aicpa.org/download/antifraud/SAS-99-Exhibit.pdf.

[Jeffery Lehman, 2004] Jeffery Lehman, S. P. (2004). West's encyclopedia of american law. Available at http://www.answers.com/topic/fraud, accessed on January 5th, 2010.

[Jin et al., 2007] Jin, R., McCallen, S., and Almaas, E. (2007). Trend motif: A graph mining approach for analysis of dynamic complex networks. *Data Mining, IEEE International Conference on*, 0:541–546.

[Jonyer et al., 2000] Jonyer, I., Holder, L. B., and Cook, D. J. (2000). Graph-based hierarchical conceptual clustering. *International Journal on Artificial Intelligence Tools*, 2:107–135.

[Katkov, 2006] Katkov, N. (2006). Evaluating the Vendors of Anti-Money Laundering Solutions 2006. Technical report, Celent. Available at http://www.celent.com/67 _69.htm, accessed on January 5th, 2010.

[Ketkar, 2005] Ketkar, N. S. (2005). Subdue: compression-based frequent pattern discovery in graph data. In *in OSDM 05: Proceedings of the 1st international workshop on open source data mining*, pages 71–76. ACM Press.

[Kim et al., 2003a] Kim, J., Ong, A., and Overill, R. E. (2003a). Computational immunology for fraud detection (cifd) final report october 2003. Available at www.dcs.kcl.ac.uk/staff/richard/CIFD _final _report.doc, accessed on January 5th, 2010.

[Kim et al., 2003b] Kim, J., Ong, A., and Overill, R. E. (2003b). Design of an artificial immune system as a novel anomaly detector for combating financial fraud in the retail sector. In *IEEE Congress on Evolutionary Computation (1)*, pages 405–412. IEEE.

[Kirkland et al., 1998] Kirkland, J. D., Senator, T. E., Hayden, J. J., Dybala, T., Goldberg, H. G., and Shyr, P. (1998). The nasd regulation advanced detection system (ads). In *AAAI/IAAI*, pages 1055–1062.

[Kirkland et al., 1999] Kirkland, J. D., Senator, T. E., Hayden, J. J., Dybala, T., Goldberg, H. G., and Shyr, P. (1999). The nasd regulation advanced-detection system (ads). *AI Magazine*, 20(1):55–67.

[Kou et al., 2004] Kou, Y., Lu, C.-T., Sirwongwattana, S., and Huang, Y.-P. (2004). Survey of fraud detection techniques. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 2, pages 749–754 Vol.2.

[Kubica et al., 2003] Kubica, J., Moore, A., Cohn, D., and Schneider, J. (2003). cgraph: A fast graph-based method for link analysis and queries. In Grobelnik, M., Milic-Frayling, N., and Mladenic, D., editors, *Proceedings of the 2003 IJCAI Text-Mining & Link-Analysis Workshop*, pages 22–31.

[Kuramochi and Karypis, 2005] Kuramochi, M. and Karypis, G. (2005). Finding frequent patterns in a large sparse graph*. *Data Min. Knowl. Discov.*, 11(3):243–271.

[Luell, 2005] Luell, J. (2005). Analytical fraud detection. Master's thesis, University of Zurich.

[Lundin et al., 2002] Lundin, E., Kvarnström, H., and Jonsson, E. (2002). A synthetic fraud data generation methodology. In Deng, R. H., Qing, S., Bao, F., and Zhou, J., editors, *ICICS*, volume 2513 of *Lecture Notes in Computer Science*, pages 265–277. Springer.

[Macskassy and Provost, 2005] Macskassy, S. A. and Provost, F. (2005). Suspicion scoring based on guilt-by-association, collective inference, and focused data access. In *International Conference on Intelligence Analysis*.

[Maes et al., 2000] Maes, S., Tuyls, K., and Vanschoenwinkel, B. (2000). Machine learning techniques for fraud detection. Master thesis, VUB, 2000., available at citeseer.ist.psu.edu/maes00machine.html, accessed on January 5th, 2010.

[Maes et al., 1993] Maes, S., Tuyls, K., Vanschoenwinkel, B., and Manderick, B. (1993). Credit card fraud detection using bayesian and neural networks. In *In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association Neurological Surgeons*, pages 261–270.

[Major and Riedinger, 2002] Major, J. A. and Riedinger, D. R. (2002). Efd: A hybrid knowledge/statistical-based system for the detection of fraud. *Journal of Risk & Insurance*.

[McKay, 1990] McKay, B. (1990). Naury's user guide. Technical report, Department of Computer Science, Australian National University, Canberra.

[Meier, 2009] Meier, M. (2009). The extended graphslider framework. Master's thesis, University of Zurich, Department of Informatics.

[Moll, 2009] Moll, L. (2009). Anti Money Laundering under real world conditions. Master's thesis, University of Zurich.

[O'Madadhain et al., 2005] O'Madadhain, J., Hutchins, J., and Smyth, P. (2005). Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30.

[Osborne, 2009] Osborne, V. (2009). Kpmg forensic fraud barometer. Available at http://www.yhff.co.uk/Fraud

[Phua et al., 2004] Phua, C., Alahakoon, D., and Lee, V. C. S. (2004). Minority report in fraud detection: classification of skewed data. *SIGKDD Explorations*, 6(1):50–59.

[Phua et al., 2005] Phua, C., Lee, V., Smith, K., and Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research.

[Pottenger et al., 2006] Pottenger, W. M., Yang, X., and Zanias, S. V. (2006). Link Analysis Survey Status Update January 2006.

[Prodromidis and Stolfo, 1999] Prodromidis, A. L. and Stolfo, S. (1999). Agent-based distributed learning applied to fraud detection. In *In Sixteenth National Conference on Artificial Intelligence*, pages 014–99.

[Sarbanes and Oxley, 2002] Sarbanes, P. S. and Oxley, M. (2002). Sarbanes-oxley act of 2002. Pub. L. No. 107-204, 116 Stat. 745 (codified as amended in scattered sections of 15 U.S.C.), available at http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3763.ENR, accessed on January 4th, 2010.

[Schneier, 2006] Schneier, B. (2006). Data mining for terrorists. Available at http://www.schneier.com/blog/archives/2006/03/data_mining_for.html, accessed on January 5th, 2010.

[Senator, 2000] Senator, T. E. (2000). Ongoing management and application of discovered knowledge in a large regulatory organization: a case study of the use and impact of nasd regulation's advanced detection system (rads). In *KDD*, pages 44–53.

[Senator et al., 1995] Senator, T. E., Goldberg, H. G., Wooton, J., Cottini, M. A., Khan, A. F. U., Klinger, C. D., Llamas, W. M., Marrone, M. P., and Wong, R. W. H. (1995). The financial crimes enforcement network ai system (fais) identifying potential money laundering from reports of large cash transactions. *AI Magazine*, 16(4):21–39.

[Shasha et al., 2002] Shasha, D., Wang, J. T. L., and Giugno, R. (2002). Algorithmics and applications of tree and graph searching. In *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 39–52, New York, NY, USA. ACM.

[Singhal, 2001] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

[Stolfo et al., 1997] Stolfo, S., Fan, W., Lee, W., Prodromidis, A., and Chan, P. (1997). Credit card fraud detection using meta-learning: Issues and initial results. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.5003, accessed on January 5th, 2010.

[Stolfo et al., 1998] Stolfo, S. J., Fan, D. W., Prodromidis, A., Lee, W., Tselepis, S., and Chan, P. K. (1998). Agent-based fraud and intrusion detection in financial information systems. In *IEEE Symposium on Security and Privacy*.

[Tong and Faloutsos, 2006] Tong, H. and Faloutsos, C. (2006). Center-piece subgraphs: problem definition and fast solutions. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 404–413, New York, NY, USA. ACM.

[Tong et al., 2007] Tong, H., Faloutsos, C., Gallagher, B., and Eliassi-Rad, T. (2007). Fast best-effort pattern matching in large attributed graphs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746, New York, NY, USA. ACM.

[Ullmann, 1976] Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42.

[Vilalta et al., 2004] Vilalta, R., Giraud-Carrier, C., Brazdil, P., and Soares, C. (2004). Using meta-learning to support data mining. *International Journal of Computer Science and Applications*, 1(1):31–45. DBLP.

[Wang and Yang, 2007] Wang, S.-N. and Yang, J.-G. (2007). A Money Laundering Risk Evaluation Method Based on Decision Tree. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 1, pages 283–286.

[Washio and Motoda, 2003] Washio, T. and Motoda, H. (2003). State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68.

[Wasserman and Faust, 1994] Wasserman, S. and Faust, K. (1994). *Social Network Analysis Methods and Applications*. Cambride University Press.

[Weiss, 2006] Weiss, D. (2006). Mining customer networks and inter-product relations in internet / digital entertainment provider data. Master's thesis, University of Zurich.

[Weston et al., 2008] Weston, D. J., Hand, D. J., Adams, N. M., Whitrow, C., and Juszczak, P. (2008). Plastic card fraud detection using peer group analysis. *Adv. Data Analysis and Classification*, 2(1):45–62.

[Whitrow et al., 2009] Whitrow, C., Hand, D. J., Juszczak, P., Weston, D. J., and Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Min. Knowl. Discov.*, 18(1):30–55.

[Wolverton et al., 2003] Wolverton, M., Berry, P., Harrison, I., Lowrance, J., Morley, D., Rodriguez, A., Ruspini, E., and Thomere, J. (2003). Law: A workbench for approximate pattern matching in relational data. In *In The Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03*, pages 143–150.

[Xu et al., 2006] Xu, J., Sung, A., and Liu, Q. (2006). Tree Based Behavior Monitoring for Adaptive Fraud Detection. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 1208–1211.

[Xu and Chen, 2005] Xu, J. J. and Chen, H. (2005). Crimenet explorer: a framework for criminal network knowledge discovery. *ACM Trans. Inf. Syst.*, 23(2):201–226.

[Yamanishi et al., 2000] Yamanishi, K., Takeuchi, J.-I., Williams, G., and Milne, P. (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 320–324, New York, NY, USA. ACM.

[Yue et al., 2007] Yue, D., Wu, X., Wang, Y., Li, Y., and Chu, C.-H. (2007). A Review of Data Mining-Based Financial Fraud Detection Research. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 5519–5522.

[Zaki et al., 2004] Zaki, M. J., De, N., Gao, F., Palmerini, P., Parimi, N., Pathuri, J., Phoophakdee, B., and Urban, J. (2004). Generic pattern mining via data mining template library. In Boulicaut, J.-F., Raedt, L. D., and Mannila, H., editors, *Constraint-Based Mining and Inductive Databases*, volume 3848 of *Lecture Notes in Computer Science*, pages 362–379. Springer.

[Zhang et al., 2003] Zhang, Z. M., Salerno, J. J., and Yu, P. S. (2003). Applying data mining in investigating money laundering crimes. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 747–752, New York, NY, USA. ACM.

[Ziegler et al., 2007] Ziegler, H., Nietzschmann, T., and Keim, D. (2007). Visual Exploration and Discovery of Atypical Behavior in Financial Time Series Data using Two-Dimensional Colormaps. In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 308–315.

B

Curriculum Vitae

Personal Information

Name / Surname	Jonas Luell
Nationality	Swiss
Date of birth	28.02.1978
Gender	male

Education

Dates	10/2005 - 03/2010
Title of qualification is going to award	Doctorate in Computer Science
Awarding University	University of Zurich, Zurich, Switzerland

Dates	10/2005 - 03/2010
Title of qualification awarded	M.Sc. in Computer Science
Awarding University	University of Zurich, Zurich, Switzerland

Work Experience

Dates	10/2005 - 12/2009
Position held	Research Assistant
Employer	Department of Informatics, University of Zurich
Dates	10/2005 - 11/2009
Position held	External Employee, Project Leader, Project Associate
Employer	”Alphafin” - Real name not given due to nondisclosure agreement
Dates	06/2003 - 09/2003
Position held	Internship position, Data analyst
Employer	kdlabs AG, Zurich
Dates	2000 - 2010
Position held	Various teaching assignments
Employer	IFA, TBZ, University of Zurich